

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-312378

(43)Date of publication of application : 09.11.1999

(51)Int.Cl. G11B 27/00

(21)Application number : 10-229704

(71)Applicant : SONY CORP

(22)Date of filing : 14.08.1998

(72)Inventor : KIMURA SATORU
ISHIKAWA AKIO

(30)Priority

Priority number : 09288178 Priority date : 21.10.1997 Priority country : JP
10 46855 27.02.1998

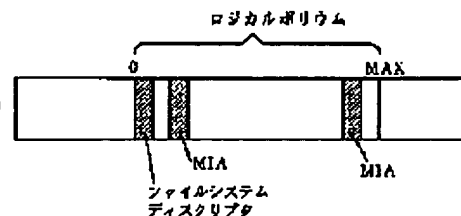
JP

(54) RECORDING AND REPRODUCING DEVICE, FILE MANAGING METHOD AND PROVIDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To obtain a file management system in which a private person can perform the recording and reproducing of AV signals with respect to a disk simply in his home by providing a first recording means recording the file of AV data and a second recording means recording management information of the file at the first place and the second place of a logical volume.

SOLUTION: In a recording and reproducing device, first and second recording means record the file of AV data and management information of the file in unit recording means MIAs of two places being in the interval 0 to WAX of a logical volume in a disk shaped recording medium. The unit of information recordable on the disk shaped recording medium is recorded in the means MIAs and the length of the unit of the information is set by a setting means and information concerning defective sectors and unused sectors are included in the recorded management information. Moreover, a computer making these respective processings to be executed provides a readable program. Thus, a private person can record or reproduce a compressed video and a compressed video signal easily in his home.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C): 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-312378

(43) 公開日 平成11年(1999)11月9日

(51) Int.Cl.⁶

G 1 1 B 27/00

識別記号

F I

G 1 1 B 27/00

D

審査請求 未請求 請求項の数10 O L (全 38 頁)

(21) 出願番号 特願平10-229704

(22) 出願日 平成10年(1998)8月14日

(31) 優先権主張番号 特願平9-288178

(32) 優先日 平9(1997)10月21日

(33) 優先権主張国 日本 (J P)

(31) 優先権主張番号 特願平10-46855

(32) 優先日 平10(1998)2月27日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 木村 哲

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72) 発明者 石川 明雄

東京都品川区北品川6丁目7番35号 ソニー株式会社内

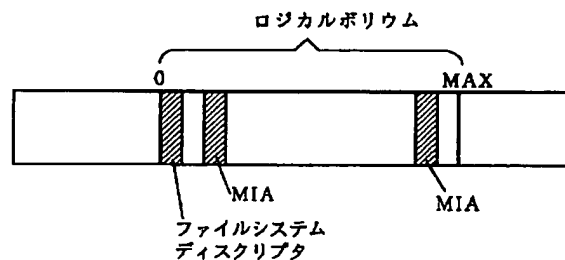
(74) 代理人 弁理士 稲本 義雄

(54) 【発明の名称】 記録再生装置、ファイル管理方法、並びに提供媒体

(57) 【要約】

【課題】 個人が家庭内で簡単にディスクにAV信号を記録再生するためのファイルシステムを実現する。

【解決手段】 ディスク状記録媒体を使用する記録再生装置のためのファイルシステムにおいて、AVデータのファイルを管理する管理情報を記録し、管理情報を、論理ボリュームの少なくとも2個所に記録する。



【特許請求の範囲】

【請求項1】 ディスク状記録媒体に対してAVデータを記録または再生する記録再生装置において、AVデータのファイルを記録する第1の記録手段と、前記ファイルの管理情報を、論理ボリウムの少なくとも2個所に記録する第2の記録手段とを備えることを特徴とする記録再生装置。

【請求項2】 前記記録手段に記録される管理情報には、少なくとも、欠陥セクタ、および未使用セクタに関する情報が含まれることを特徴とする請求項1記載の記録再生装置。

【請求項3】 ディスク状記録媒体に対してAVデータを記録または再生する記録再生装置のファイル管理方法において、

AVデータのファイルを記録する第1の記録ステップと、前記ファイルの管理情報を、論理ボリウムの少なくとも2個所に記録する第2の記録ステップとを含むことを特徴とするファイル管理方法。

【請求項4】 ディスク状記録媒体に対してAVデータを記録または再生する記録再生装置に、AVデータのファイルを記録する第1の記録ステップと、前記ファイルの管理情報を、論理ボリウムの少なくとも2個所に記録する第2の記録ステップとを含む処理を実行させるコンピュータが読みとり可能なプログラムを提供することを特徴とする提供媒体。

【請求項5】 ディスク状記録媒体に対して記録可能な情報の単位を記録する単位記録手段と、前記単位記録手段により記録する情報の単位の長さを設定する設定手段と、前記ディスク状記録媒体に記録されるファイルに対応して、前記設定手段により設定された長さの単位のうち、そのファイルを記録するとき採用された前記単位の長さを識別する識別情報を記録する識別情報記録手段とを備えることを特徴とする記録再生装置。

【請求項6】 前記設定手段は、AVデータの単位の長さを、コンピュータデータの単位の長さより長く設定することを特徴とする請求項5に記載の記録再生装置。

【請求項7】 前記ディスク状記録媒体を複数のブロックに分割し、分割された前記ブロックの1/2以上の領域にデータが記録されるように制御する制御手段をさらに備えることを特徴とする請求項5に記載の記録再生装置。

【請求項8】 前記ディスク状記録媒体に情報を記録するトラックを複数のブロックに分割し、分割された前記ブロックの(n-1)/n以上の領域にデータが記録されるように制御する制御手段をさらに備えることを特徴とする請求項5に記載の記録再生装置。

【請求項9】 ディスク状記録媒体に対して情報を記録または再生する記録再生装置のファイル管理方法において、

ディスク状記録媒体に対して記録可能な情報の単位を記録する単位記録ステップと、

前記単位記録ステップで記録する情報の単位の長さを設定する設定ステップと、

前記ディスク状記録媒体に記録されるファイルに対応して、前記設定ステップで設定された長さの単位のうち、そのファイルを記録するとき採用された前記単位の長さを識別する識別情報を記録する識別情報記録ステップとを含むことを特徴とするファイル管理方法。

【請求項10】 ディスク状記録媒体に対して情報を記録または再生する記録再生装置に、

前記ディスク状記録媒体に対して記録可能な情報の単位を記録する単位記録ステップと、

前記単位記録ステップで記録する情報の単位の長さを設定する設定ステップと、

前記ディスク状記録媒体に記録されるファイルに対応して、前記設定ステップで設定された長さの単位のうち、そのファイルを記録するとき採用された前記単位の長さを識別する識別情報を記録する識別情報記録ステップとを含む処理を実行させるコンピュータが読みとり可能なプログラムを提供することを特徴とする提供媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、記録再生装置、ファイル管理方法、並びに提供媒体に関し、特に、ディスク状記録媒体を使用する記録再生装置（VDR: Video Disc Recorder）に使用するファイルシステムに関する。

【0002】

【従来の技術】ディスク状記録媒体にデータを記録するためのファイルシステムとしては、ISO/IEC13346:1995, "Information technology - Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange."が知られている。このファイルシステムは、各種データを記録するための汎用的なファイルシステムであり、個人が家庭内でディスクに圧縮されたデジタルAV（音声、ビデオ）信号を記録するためのものではない。従って、圧縮されたデジタルAV（音声、ビデオ）信号を記録するには必ずしも十分なものではない。従って、AV信号を記録するに最適なファイルシステム、及びボリウムが求められている。

【0003】

【発明が解決しようとする課題】従って、個人が家庭内で簡単にディスクにAV信号を記録再生するためのファイルシステムが必要である。

【0004】

【課題を解決するための手段】請求項1に記載の記録再生装置は、AVデータのファイルを記録する第1の記録手段と、管理情報を、論理ボリウムの少なくとも2個所に記録する第2の記録手段とを備えることを特徴とする。

【0005】請求項3に記載のファイル管理方法は、AVデータのファイルを記録する第1の記録ステップと、管理情報を、論理ボリュームの少なくとも2個所に記録する第2の記録ステップとを含むことを特徴とする。

【0006】請求項4に記載の提供媒体は、AVデータのファイルを記録する第1の記録ステップと、管理情報を、論理ボリュームの少なくとも2個所に記録する第2の記録ステップとを含む処理を実行させるコンピュータが読みとり可能なプログラムを提供することを特徴とする。

【0007】請求項5に記載の記録再生装置は、ディスク状記録媒体に対して記録可能な情報の単位を記録する単位記録手段と、単位記録手段により記録する情報の単位の長さを設定する設定手段と、ディスク状記録媒体に記録されるファイルに対応して、設定手段により設定された長さの単位のうち、そのファイルを記録するとき採用された単位の長さを識別する識別情報を記録する識別情報記録手段とを備えることを特徴とする。

【0008】請求項9に記載のファイル管理方法は、ディスク状記録媒体に対して記録可能な情報の単位を記録する単位記録ステップと、単位記録ステップで記録する情報の単位の長さを設定する設定ステップと、ディスク状記録媒体に記録されるファイルに対応して、設定ステップで設定された長さの単位のうち、そのファイルを記録するとき採用された単位の長さを識別する識別情報を記録する識別情報記録ステップとを含むことを特徴とする。

【0009】請求項10に記載の提供媒体は、ディスク状記録媒体に対して記録可能な情報の単位を記録する単位記録ステップと、単位記録ステップで記録する情報の単位の長さを設定する設定ステップと、ディスク状記録媒体に記録されるファイルに対応して、設定ステップで設定された長さの単位のうち、そのファイルを記録するとき採用された単位の長さを識別する識別情報を記録する識別情報記録ステップとを含む処理を実行させるコンピュータが読みとり可能なプログラムを提供することを特徴とする。

【0010】請求項1に記載の記録再生装置、請求項3に記載のファイル管理方法、および請求項4に記載の提供媒体においては、AVデータのファイルを記録し、管理情報を、論理ボリュームの少なくとも2個所に記録する。

【0011】請求項5に記載の記録再生装置、請求項9に記載のファイル管理方法、および請求項10に記載の提供媒体においては、ディスク状記録媒体に対して記録可能な情報の単位を記録し、記録する情報の単位の長さを設定し、ディスク状記録媒体に記録されるファイルに対応して、設定された長さの単位のうち、そのファイルを記録するとき採用された単位の長さを識別する。

【0012】

【発明の実施の形態】以下に本発明の実施の形態を説明

するが、特許請求の範囲に記載の発明の各手段と以下の実施の形態との対応関係を明らかにするために、各手段の後の括弧内に、対応する実施の形態（但し一例）を付加して本発明の特徴を記述すると、次のようになる。但し勿論この記載は、各手段を記載したものに限定することを意味するものではない。

【0013】すなわち、請求項1に記載の記録再生装置は、AVデータのファイルを記録する第1の記録手段（例えば、図19のドライブ部7）と、管理情報を、論理ボリュームの少なくとも2個所に記録する第2の記録手段（例えば、図19のドライブ部7）とを備えることを特徴とする。

【0014】請求項5に記載の記録再生装置は、ディスク状記録媒体に対して記録可能な情報の単位を記録する単位記録手段（例えば、図3のMIA）と、単位記録手段により記録する情報の単位の長さを設定する設定手段（例えば、図4の処理のステップS11）と、ディスク状記録媒体に記録されるファイルに対応して、設定手段により設定された長さの単位のうち、そのファイルを記録するとき採用された単位の長さを識別する識別情報を記録する識別情報記録手段（例えば、図6の処理のステップS23）とを備えることを特徴とする。

【0015】請求項7に記載の記録再生装置は、前記ディスク状記録媒体を複数のブロックに分割し、分割された前記ブロックの1/2以上の領域にデータが記録されるように制御する制御手段（例えば、図19のファイル管理部6）をさらに備えることを特徴とする。

【0016】はじめに、ディスク状記録媒体のフォーマット方式について説明する。図1は、ディスク状記録媒体全体のフォーマットを説明する図である。ディスクは、長さが可変長の複数のアロケーションエクステントに分割される。アロケーションエクステントは、長さが固定の複数のブロックから構成される。ブロックは、所定の数の物理セクタから構成される。

【0017】図2は、アンカーディスクリプタについて説明する図である。ディスク内に、4つのアンカーディスクリプタが配置される。アンカーディスクリプタには、ボリューム管理用マネージメントインフォメーションエリアの位置が記録されている。ボリューム管理用マネージメントインフォメーションエリアのボリュームストラクチャディスクリプタには、フィジカルボリュームインフォメーション、パーテーションインフォメーション、ロジカルボリュームインフォメーション、およびパーテーションマップが含まれている。

【0018】ボリュームストラクチャディスクリプタには、ユーザエリアとしてのロジカルボリュームが記述されている。図3は、ロジカルボリュームを説明する図である。ロジカルボリュームには、ファイルシステムディスクリプタが、配置されている。ロジカルボリュームの先頭付近、および終了付近には、それぞれ、MIA(Management I

nformation Area)が配置されている。MIAには、ファイルテーブル、アロケーションエクステンテーブル、アロケーションストラテジテーブル、ディフェクトインフォメーションテーブル、エクステンデッドアトリビュートテーブルが含まれている。アロケーションエクステンツの長さは、アロケーションストラテジテーブルを構成するアロケーションストラテジレコードに記述される。

【0019】ユーザは、ディスクにファイルのデータを記録する前に、そのディスクに記録するデータのアロケーションエクステンツの長さを予め設定する。これにより、例えば、AVデータは、より長い長さのアロケーションエクステンツのフォーマットで記録し、PCデータは短い長さのアロケーションエクステンツのフォーマットで記録することが可能となる。AVデータは連続するデータであることが多いので、アロケーションエクステンツの長さを長くした方が、データをより効率的に記録再生することができる。

【0020】図4は、アロケーションエクステンツの長さの設定の処理を説明するフローチャートである。ステップS11において、後述するドライブ部7は、ユーザからの設定入力に対応して、MIAに含まれるアロケーションストラテジテーブルに、アロケーションエクステンツの設定された長さに対応したアロケーションストラテジレコードを書き込む。アロケーションストラテジテーブルには、複数のアロケーションストラテジレコードを書き込むことができる。図5は、アロケーションエクステンツの長さをユーザが設定する画面の例を示す図である。アロケーションエクステンツの長さとしては、4MByte以上、64KByte、2kByteなど任意の長さが設定可能であり、かつ、複数の長さの設定が可能である。そのディスクには、予め設定した長さのアロケーションエクステンツのフォーマットの中から指定されたもののみ記録が可能である。

【0021】このように、アロケーションエクステンツの長さを設定し、ディスクに記録した後、そのディスクにデータを記録する場合の処理は、図6のフローチャー

トに示すようになる。ステップS21において、ユーザは、これから記録するデータのアロケーションエクステンツの長さを選択する。図7は、アロケーションエクステンツの長さを選択する画面の例を表している。この長さとしては、そのディスクに予め設定された値だけが表示される。画面のボタンを操作することで、ボタンに対応するアロケーションエクステンツの長さが、選択される。AVデータを記録するとき、PCデータを記録するときに較べて、より長いアロケーションエクステンツを指定することで、より効率的なデータの記録が可能になる。アロケーションエクステンツの長さの指定により、アロケーションストラテジテーブル内に配置されたアロケーションストラテジレコードが指定される。指定が完了すると、ステップS22において、ドライブ部7は、入力されたデータをディスクに記録する。データの記録が完了すると、ステップS23において、ドライブ部7は、ディスクにそのファイルのアロケーションエクステンツの長さに対応した番号を記録する。後述するファイル管理部6は、アロケーションエクステンツの長さに対応した番号を知ることにより、対応するアロケーションストラテジレコードの内容を利用することができる。

【0022】後述する図19のシステムコントロール部5が、AVデータを記録しようとしているのか、PCデータを記録しようとしているのかを判断することができる場合には、前述のステップS21をユーザからの入力なしに行うことも可能である。

【0023】以上のように、ディスクにファイルが記録される。

【0024】ポリウムの構成について説明する。ディスクエクステンツ(DescExtent)は、後述のMIA内に記録されたディスクリプタ(descriptor)中の後述するMIB(Management Information Block)にアライメントされた領域を表現するのに用いられる。ディスクエクステンツは表1に示す様式で記録する。

【0025】

【表1】

Desc Extent			
RBP	Length	Name	Contents
0	2	Offset (Number of MIB) from top of a descriptor	Uint 16
2	2	Length (Number of MIB)	Uint 16

【0026】オフセットフロムトップオブディスクリプタ(Offset from top of a descriptor:RBP 0)は、ディスクリプタ(記述子)の先頭MIBから領域までのオフセット(MIB数)を指定する。レンジス(Length :RBP 2)は領域の大きさ(MIB数)を指定する。

【0027】PDLエントリ(Primary Defect List Entry)は、ディフェクトマネージメント(defect management

1)においてスリッピング(slipping)を行う物理セクタ(physical sector)の物理セクタサイズ(physical sector size)を記録するのに用いる。PDLエントリは表2に示す様式で記録する。

【0028】

【表2】

PDL Entry

RBP	Length	Name	Contents
0	4	Physical Sector Number of Defect Sector	Unit 32

【0029】フィジカルセクタナンバオブディフェクトセクタ(Physical Sector Number of Defect Sector :RBP 0)はスリッピングを行う物理セクタの物理セクタ番号を指定する。

【0030】SDLエン트리(Secondary Defect List Entry)は、ディフェクトマネージメントにおいてリニアリブ

SDL Entry

RBP	Length	Name	Contents
0	4	Physical Sector Number of Defect Sector	Unit 32
4	4	Physical Sector Number of Spare Sector	Unit 32

【0032】表3のフィジカルセクタナンバオブディフェクトセクタはリニアリブレスメントを行う物理セクタの物理セクタ番号を指定する。フィジカルセクタナンバオブスペアセクタ(Physical Sector Number of Spare Sector:RBP 4)はリニアリブレスメントで使用する代替物理セクタの物理セクタ番号を指定する。

【0033】アンカーポイント(Anchor points)は、ボリューム構造解析の開始点である。アンカーポイントにはアンカーディスクリプタ(Anchor Descriptor)が記録される。アンカーポイントである物理セクタの物理セクタ番号は規定しない。

【0034】ただし、VDRでは、以下のように規定される。すなわちROM(Read Only Memory)ディスク、RAM(Random Access Memory)ディスクの場合はCh, 20h, LPSN(Last Physical Sector Number)-20h, LPSN-Ch(hが最後に付いた数値は、16進数を表す)をアンカーポイントとする。パーシャルROMディスクの場合には、ROM、RAMそれぞれの領域でのCh, 20h, LPSN-20h, LPSN-Chをアンカーポイントとする。この場合、もしRAM領域のアンカーポイントに適切な情報が記録されている場合にはそれを使用し、適切な情報が記録されていない場合には、ROM領域の情報を使用する。

【0035】アンカーディスクリプタはアンカーポイントである物理セクタにバイト位置0から記録される。アンカーディスクリプタの大きさは物理セクタサイズ以下である。また、ディスクリプタの最終バイトの次のバイトからその物理セクタの最後のバイトまでの領域は、将来の拡張の為に予約されており、全てのバイトに#00を設定する。アンカーディスクリプタには、メイン(Main) MIA領域の定義とリザーブ(Reserve) MIA領域の定義、そしてそれぞれのMIAマップ(Map)の位置情報などが記録される。

【0036】ボリュームに関する各種の情報はボリューム管

理用マネージメントインフォメーションエリア(MIA)に記録される。信頼性確保のため、等しい内容の情報を持つMIAが物理ボリューム上の2ヶ所に記録され、それぞれメインMIA、リザーブMIAと称する。MIA内の物理セクタはマネージメントインフォメーションブロック(Management Information Block:MIB)と呼ばれ、その物理セクタ番号のMIAの先頭MIBからのオフセットはマネージメントインフォメーションブロック番号(Management Information Block Number :MIB Number)と称する。MIBの指定にはMIB番号が使われる。MIAは、欠陥などにより使用することが出来ないMIB、未使用のMIB、並びにメインMIAのMIAマップ(MIA Map for Main MIA)、リザーブMIAのMIAマップ(MIA Map for Reserve MIA)、ボリュームストラクチャディスクリプタ(Volume Structure Descriptor)、メディアインフォメーションディスクリプタ(Media Information Descriptor)、ドライブインフォメーションディスクリプタ(Drive Information Descriptor)、およびエクステンデッドデータディスクリプタ(Extended Data Descriptor)のデータを記録するのに使われるMIBから構成される。

【0031】

【表3】

理用マネージメントインフォメーションエリア(MIA)に記録される。信頼性確保のため、等しい内容の情報を持つMIAが物理ボリューム上の2ヶ所に記録され、それぞれメインMIA、リザーブMIAと称する。MIA内の物理セクタはマネージメントインフォメーションブロック(Management Information Block:MIB)と呼ばれ、その物理セクタ番号のMIAの先頭MIBからのオフセットはマネージメントインフォメーションブロック番号(Management Information Block Number :MIB Number)と称する。MIBの指定にはMIB番号が使われる。MIAは、欠陥などにより使用することが出来ないMIB、未使用のMIB、並びにメインMIAのMIAマップ(MIA Map for Main MIA)、リザーブMIAのMIAマップ(MIA Map for Reserve MIA)、ボリュームストラクチャディスクリプタ(Volume Structure Descriptor)、メディアインフォメーションディスクリプタ(Media Information Descriptor)、ドライブインフォメーションディスクリプタ(Drive Information Descriptor)、およびエクステンデッドデータディスクリプタ(Extended Data Descriptor)のデータを記録するのに使われるMIBから構成される。

【0037】MIA中のMIBがどの目的で使われているかはMIAマップに記録される。メインMIAとリザーブMIAの開始位置と大きさ、MIA中のMIAマップの位置はアンカーディスクリプタで規定される。上記のデータは、1つのMIB内に記録される場合、または複数のMIBにわたって記録される場合がある。データが複数のMIBに記録される場合、どのMIBをどの順番で連結するかはMIAマップ中のMapエン트리(Map Entries)フィールドに記録される。データがMIBの途中で終わった場合には、データの終わりの次のバイトからそのMIBの最後のバイトまでは、#00を設定する。

【0038】つぎに、区分(Partition)について説明する。ボリュームストラクチャディスクリプタ(Volume S

structure Descriptor) 中のパーティションインフォメーション (Partition Information) で定義されるデータ記憶領域をパーティション (partition) と称する。一つの物理ボリュームを複数のパーティションに分けることができる。物理ボリューム内でパーティションを特定するための番号をパーティション番号と称する。パーティション番号は0から始まり単調に1ずつ増加する整数である。同一のパーティション内の物理セクタは全て同じ物理セクタサイズである。

【0039】パーティションは、ボリュームストラクチャディスクリプタの中にパーティションインフォメーション (Partition Information) の表として定義する。パーティションインフォメーションは、パーティションの先頭の物理セクタの物理セクタ番号とそのパーティションに属する物理セクタの数でパーティションを定義する。物理ボリューム中には必ず一つ以上のパーティションが定義される。パーティション番号は、パーティションインフォメーションがボリュームストラクチャディスクリプタに記録された順序で決定される。1番目のパーティションインフォメーションで定義されるパーティションのパーティション番号は0であり、2番目は1であり、以降1ずつ増え、n番目はn-1である。

【0040】つぎに論理ボリューム (Logical volume) について説明する。論理ボリュームとは、ボリュームストラクチャディスクリプタの論理ボリュームインフォメーション (Logical Volume Information) において、パーティションの集まりとして定義されるデータ記憶領域をいう。論理ボリュームの領域は、論理ボリュームインフォメーションのパーティションマップ (Partition Map) の記述順にパーティション領域を連結して構成される。パーティションマップは、物理ボリュームを一意に定めるボリュームアイデンティファイア (Volume Identifier) とその物理ボリュームでのパーティション番号の組で論理ボリュームに属するパーティションを指定する。論理ボリュームは、異なる物理ボリュームに属するパーティションから構成されてもよく、1つのパーティションが複数の論理ボリュームに属していてもよい。

【0041】論理ボリュームはパーティションの区切れ目や物理セクタなどに関係なく1つの領域として扱われ、その内容は論理セクタ単位に読み書きされる。論理セクタ番号は0から始まり単調に1ずつ増加する整数である。論理ボリュームの大きさが論理セクタサイズの倍数でない場合、最終物理セクタに生ずる半端な領域は、将来の拡張の為に予約されており使用しない。ボリュームストラクチャディスクリプタは、その物理ボリュームに含まれるパーティションに関する情報の定義や論理ボリュームの定義などが記述される。複数の物理ボリュームにまたがる論理ボリュームを定義する場合、必ずパーティション番号0のパーティションが定義されている物理ボリュームのボリュームストラクチャディスクリプタに、論理ボリュームインフ

ォメーションが記述される。

【0042】なお、信頼性確保の為、パーティション番号0以外のパーティションが属する物理ボリュームのボリュームストラクチャディスクリプタに論理ボリュームインフォメーションを記述してもよい。ボリュームストラクチャディスクリプタは、MIAに記録される。

【0043】つぎにディフェクトマネージメント (Defect management) について、説明する。各パーティション毎に、スリッピングとリニアリプレースメントによるディフェクトマネージメントが可能である。それぞれのパーティションに対しディフェクトマネージメントを行うか否かの指定は、ボリュームストラクチャディスクリプタのパーティションインフォメーションで行う。スリッピングとリニアリプレースメントの為に用いる代替データ領域をスペアエリア (spare area) と呼ぶ。ディフェクトマネージメントを行うパーティションと同一の論理ボリュームに属するパーティション内には、必ず1つの以上のスペアエリアを確保する。また、リニアリプレースメントを行う場合、そのパーティション領域の最後は、スペアエリアとなる。

【0044】スリッピングを行う場合、そのパーティション領域の最後に確保されたスペアエリアの先頭部分は、スペアエリアとして使用する。また、リニアリプレースメントを行う場合、代替データ領域は、同一の論理ボリュームに属し、かつ、同一の物理ボリュームに属するパーティションであれば、ディフェクトセクタ (defect sector) のあるパーティション内のスペアエリア以外のスペアエリアを使用しても良い。

【0045】スリッピングとリニアリプレースメントに関する情報は、ボリュームストラクチャディスクリプタのディフェクトリストインフォメーション (Defect List Information) に記録される。スリッピングに関する情報は、プライマリディフェクトリスト (Primary Defect List) に、リニアリプレースメントに関する情報はセコンダリディフェクトリスト (Secondary Defect List) に記録される。

【0046】メディアに関する情報を記録する領域である、メディアインフォメーションディスクリプタ (Media Information Descriptor) は、ゾーンに関する情報などを記録する。ドライブインフォメーションディスクリプタ (Drive Information Descriptor) は、ドライブ (メディアにデータの記録再生を行う装置) に関する情報を記録する領域である。ここには、固定ドライブの場合に各種情報を記録する。

【0047】拡張データディスクリプタ (Extended Data Descriptor) は、物理ボリュームインフォメーション、パーティションインフォメーション、および論理ボリュームインフォメーションヘッダの中に記録しきれなかった拡張情報を記録する。

【0048】つぎに、ボリュームデータストラクチャ (Vo

lume data structures) について説明する。アンカーデ
 スクリプタ (Anchor Descriptor) の大きさは物理セ
 クタサイズ以下で、表4に示す様式で記録される。

【0049】
 【表4】

Anchor Descriptor

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Start Physical Sector Number of Main MIA	Uint32
12	4	Number of Physical Sectors in Main MIA	Uint32
16	4	Start Physical Sector Number of Reserve MIA	Uint32
20	4	Number of Physical Sectors in Reserve MIA	Uint32
24	2	Number of MIBs for MIA Map in Main MIA ($-x_1$)	Uint16
26	2	Number of MIBs for MIA Map in Reserve MIA ($-x_2$)	Uint16
28	$2x_1$	MIB Numbers of MIA Map for Main MIA in Main MIA	Uint16
$28+2x_1$	$2x_2$	MIB Numbers of MIA Map for Reserve MIA in Main MIA	Uint16
$28+2x_1+2x_2$	$2x_1$	MIB Numbers of MIA Map for Main MIA in Reserve MIA	Uint16
$28+4x_1+2x_2$	$2x_2$	Mib Numbers of MIA Map for Reserve MIA in Reserve MIA	Uint16

【0050】シグニチャ (Signature:BP 0) のデータ
 タイプフィールドは、16が設定される。スタートフィジ
 カルセクタナンバオブメインMIA(Start Physical Sector
 Number of Main MIA:BP 8)は、メインMIAの先頭の物理
 セクタの物理セクタ番号を指定する。ナンバオブフィジ
 カルセクタインメインMIA(Number of Physical Sectors
 in Main MIA:BP 12) は、メインMIAの物理セクタの
 数を指定する。スタートフィジカルセクタナンバオブ
 リザーブMIA(Start Physical Sector Number of Reserve
 MIA:BP 16) は、リザーブMIAの先頭の物理セクタの
 物理セクタ番号を指定する。ナンバオブフィジカルセ
 クタインリザーブMIA(Number of Physical Sectors in
 Reserve MIA :BP 20) はリザーブMIAの物理セクタの
 数を指定する。ナンバオブMIBsフォアMIAマップ
 インメインMIA(Number of MIBs for MIA Map in Main
 MIA :BP 24) は、メインMIAのMIAマップの大き
 さ(MIBの数)を指定する。ナンバオブMIBsフォ
 アMIAマップインリザーブMIA(Number of MIBs
 for MIA Map in Reserve MIA:BP 26) は、リ
 ザーブMIAのMIAマップの大きさ(MIBの数)を指
 定する。MIBナンバオブMIAマップフォアメイン
 MIAインメインMIA(MIB Numbers of MIA Map for
 Main MIA in Main MIA:BP 28) は、メインMIA
 に対するMIAマップを記録しているメインMIA中
 のMIBを指定する。MIAマップを構成するMIBの
 MIB番号は、順に設定される。

【0051】MIBナンバオブMIAマップフォアリ
 ザーブMIAインメインMIA(MIB Numbers of MIA
 Map for Reserve MIA in Main MIA :BP 28+2x1) は、
 リザーブMIAに対するMIAマップを記録している
 メインMIA中のMIBを指定する。MIAマップを
 構成するMIBのMIB番号は、順に設定される。
 MIBナンバオブMIAマップフォアメインMIA
 インリザーブMIA(MIB Numbers of MIA Map for
 Main MIA in Reserve MIA :BP 28+2x1+2x2) は、
 メインMIAに対するMIAマップを記録している
 リザーブMIA中のMIBを指定する。MIAマップ
 を構成するMIBのMIB番号は、順に設定され
 る。MIBナンバオブMIAマップフォアリ
 ザーブMIAインリザーブMIA(MIB Numbers of
 MIA Map for Reserve MIA in Reserve MIA:BP
 28+4x1+2x2) は、リザーブMIAに対する
 MIAマップを記録しているリザーブMIA中の
 MIBを指定する。MIAマップを構成するMIBの
 MIB番号は、順に設定される。

【0052】MIAマップ (MIA Map) は、MIBの
 使用状況を示すのに使われる。MIAマップは、
 各種のデータの記録に使われているMIB、欠
 陥などにより使用することが出来ないMIB、未
 使用のMIBの位置を示す。MIAマップは表5
 に示す様式で記録する。

【0053】
 【表5】

MIA Map

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Location of MIA Map	Uint16
10	2	Location of Volume Structure Descriptor	Uint16
12	2	Location of Media Information Descriptor	Uint16
14	2	Location of Drive Information Descriptor	Uint16
16	2	Location of Extended Data Descriptor	Uint16
18	2	Number of Map Entries (=n)	Uint16
20	2x1	Map Entries	bytes

【0054】シグネチャ(Signature:BP 0)のデータタイプフィールドは、2が設定される。ロケーションオブMIAマップ(Location of MIA Map:BP 8)は、MIAマップの先頭MIBのMIB番号を指定する。ロケーションオブボリュームストラクチャディスクリプタ(Location of Volume Structure Descriptor:BP 10)は、ボリュームストラクチャディスクリプタの先頭MIBのMIB番号を指定する。ロケーションオブメディアインフォメーションディスクリプタ(Location of Media Information Descriptor:BP 12)は、メディアインフォメーションディスクリプタの先頭MIBのMIB番号を指定する。ロケーションオブドライブインフォメーションディスクリプタ(Location of Drive Information Descriptor:BP 14)は、ドライブインフォメーションディスクリプタの先頭MIBのMIB番号を指定する。

【0055】ロケーションオブエクステンデッドデータディスクリプタ(Location of Extended Data Descriptor:BP 16)は、エクステンデッドデータディスクリプタの先頭MIBのMIB番号を指定する。ナンバーオブマップエントリーズ(Number of Map Entries:BP 18)は、BP 20から始まるMap Entryのエントリ数を指定する。この数は、MIA内に存在するMIBの数に等しく、#FFF0以下である。マップエントリーズ(Map Entries:BP 20)は、MIBの使用状況を指定する。1つのMap Entryは、Uint16からなっており、最初のマップエントリは最初のMIB、2番目のマップエントリは2番目のMIB、..., n番目のマップエントリはn番目のMIBに対応する。表6は、マップエントリの値を示す表である。

【0056】

【表6】

MIA Map Entry

Value	Interpretation
#0000-#FFEF	Next MIB Number
#FFF0	Unusable MIB
#FFF1	Unused MIB
#FFF2-#FFFE	Reserved
#FFFF	Last MIB of the data structure

【0057】図8は、ボリュームストラクチャディスクリプタ (Volume Structure Descriptor) の構造を示す図である。ここで、@APStは、アライントゥフィジカルセクタ (Align to Physical Sector)を示し、そのデータは、物理セクタにアライメントすることを示す。また、アライメントに際し、直前に記録すべきデータが実際に記録された場所の次のバイトからそのセクタの終りまでの領

域は、#00が設定される。

【0058】ボリュームストラクチャヘッダ (Volume Structure Descriptor Header) は、表7に従って記録される。

【0059】

【表7】

Volume Structure Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	Uint16
10	2	Reserved	#00 bytes
12	4	Offset to Physical Volume Information (=48)	Uint32
16	4	Offset to Partition Information (=416)	Uint32
20	4	Offset to Spare Area Information	Uint32
24	4	Offset to Logical Volume Information	Uint32
28	4	Offset to Defect List Information	Uint32

【0060】シグネチャ(Signature:BP 0)のデータタイプフィールドは、17が設定される。ディスクリプタサイズ(Descriptor Size:BP 8)は、ポリウムストラクチャディスクリプタの大きさ(MIB数)を指定する。リザーブド(Reserved:BP 10)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。オフセットトゥフィジカルボリュームインフォメーション(Offset to Physical Volume Information:RBP 12)は、物理ポリウムインフォメーションのポリウムストラクチャディスクリプタの先頭バイトからのオフセット(バイト数)を指定し、48を設定する。オフセットトゥパーティションインフォメーション(Offset to Partition Information:RBP 16)は、パーティションインフォメーションのポリウムストラクチャディスクリプタの先頭バイトからのオフセット(バイト数)を指定し、416を設定する。オフセットスベアエリアインフォメーション(Offset to Spare Area Information:RBP 20)は、スベアエリアインフォメーション

のポリウムストラクチャディスクリプタの先頭バイトからのオフセット(バイト数)を指定する。オフセットトゥロジカルボリュームインフォメーション(Offset to Logical Volume Information:RBP 24)は、論理ポリウムインフォメーションのポリウムストラクチャディスクリプタの先頭バイトからのオフセット(バイト数)を指定する。オフセットトゥディフェクトリストインフォメーション(Offset to Defect List Information:RBP 28)は、ディフェクトリストインフォメーションのポリウムストラクチャディスクリプタの先頭バイトからのオフセット(バイト数)を指定する。

【0061】物理ポリウムインフォメーション(Physical Volume Information)は表8に従って記録しなければならない。

【0062】

【表8】

Physical Volume Information

RBP	Length	Name	Contents
0	2	Charactor Set	Charactor Set
2	2	Physical Volume Name Size	Uint16
4	256	Physical Volume Name	bytes
260	20	Physical Volume Identifier	bytes
280	6	Creation Time	Time Stamp
286	6	Modification Time	Time Stamp
292	2	Number of Partitions (=Np)	Uint16
294	2	Number of Spare Areas (=Ns)	Uint16
296	2	Number of Partitions with Defect Management (=Ndump)	Uint16
298	2	Number of Logical Volume (=Nv)	Uint16
300	2	Reserved	#00 bytes
302	2	Extended Data Identifier	Uint16 bytes
304	64	Extended Data	bytes

【0063】キャラクタセット(Charactor Set:RBP 0)は、物理ポリウム名前フィールドに記録された物理ポリウムの名前の文字コードを指定する。フィジカルボリューム名前サイズ(Physical Volume Name Size:RBP 2)は、物理ポリウム名前フィールドに記録された物理ポリウムの名前の大きさ(バイト数)を指定する。フィジカルボリューム名前(Physical Volume Name:RBP 4)は、物理ポリウムの名前を指定する。フィジカルボリュームアイデンティファイア(Physical Volume Identifier:RBP 260)は、物理ポリウムを実用上一意に定める

ためのバイト列を指定する。クリエーションタイム(Creation Time:RBP 280)は、この物理ポリウムのボリューム構造が初めて定義された日時を指定する。モディティフィケーションタイム(Modification Time:RBP 286)は、この物理ポリウムのボリューム構造が変更された最新の日時を指定する。ナンバオブパーティション(Number of Partitions:RBP 292)は、この物理ポリウムに含まれるパーティションの数を指定し、パーティションインフォ

メーションの数と一致する。

【0064】ナンバオブスペアエリア (Number of Spare Areas:RBP 294) は、この物理ボリュームに含まれるスペアエリアの数を指定し、スペアエリアインフォメーションの数と一致する。ナンバオブパーティションウィズディフェクトマネジメント (Number of Partitions with Defect Management:RBP 296) は、この物理ボリュームに含まれるパーティションのうち、ディフェクトマネジメントを行うパーティションの数を指定し、ディフェクトリストの数と一致する。ナンバオブロジカルボリューム (Number of Logical Volumes :RBP 298) は、この物理ボリュームに含まれるパーティションが属する論理ボリュームの数を指定し、論理ボリュームインフォメーションの数と一致する。リザーブ土 (Reserved:RBP 300) は、将来の

拡張の為に予約され、全てのバイトに#00を設定する。

エクステンデッドデータアイデンティファイア (Extended Data Identifier:RBP 302) は、エクステンデッドデータフィールド、エクステンデッドデータエリアに記録されているエクステンデッドデータを特定するためのIDを指定する。エクステンデッドデータ (Extended Data:RBP 304) は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0065】パーティションインフォメーション (Partition Information) は、表9で示す様式で記録しなければならない。

【0066】

【表9】

Partition Information

RBP	Length	Name	Contents
0	4	Start Physical Sector Number	Uint32
4	4	Number of Physical Sectors	Uint16
8	4	Number of Usable Sectors	Uint16
12	4	Physical Sector size (=PSS)	Uint16
16	1	Access Type	Uint8
17	1	Usage Information	Uint8
18	2	Reserved	# 00 bytes
20	4	Location of Primary Defect List	Desc Extent
24	4	Location of secondary Defect List	Desc Extent
28	2	Reserved	# 00 bytes
30	2	Extended Data Identifier	Uint16
32	64	Extended Data	# 00 bytes

【0067】スタートフィジカルセクタナンバ (Start Physical Sector Number:RBP 0) は、パーティションを構成する領域の先頭の物理セクタの物理セクタ番号を指定する。ナンバオブフィジカルセクタズ (Number of Physical Sectors:RBP 4) は、パーティションを構成する領域の物理セクタの数を指定する。ナンバオブユーザブルセクタズ (Number of Usable Sectors:RBP 8) は、パーティションを構成する領域のうち、使用することが出来る物理セクタの総数を指定し、パーティションの全領

域からそのパーティション領域に含まれるスペアエリアを除いた領域の物理セクタの数と一致する。フィジカルセクタサイズ (Physical Sector Size:RBP 12) は、パーティションを構成する領域の物理セクタの大きさ (bytes 数) を指定する。アクセスタイプ (Access Type:RBP 16) は、このパーティションの記録特性の状態を指定する。表10は、アクセスタイプの内容を示す表である。

【0068】

【表10】

Access Type

Value	Name	Interpretation
0	Read Only	The user may not write any data in this partition
1	Write Once	The user can write data but once in this partition
2	Rewritable	The user can write data many times in this partition
3-15	Reserved	Reserved for futurer use

【0069】ユーゼジインフォメーション (Usage Information:RBP 17) は、このパーティションの利用状態を指定する。表11は、ユーゼジインフォメーションの内容

を示す表である。

【0070】

【表11】

Usage information

Bit	Interpretation
0	Used (1 : used, 0 : not used)
1	Defect management : Slipping (1 : on, 0 : off)
2	Defect management : Linear replacement (1 : on, 0 : off)
3-7	Reserved

【0071】リザーブド(Reserved:RBP 18)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。ロケーションオブプライマリディフェクトリスト(Location of Primary Defect List:RBP 20)は、このパーティションでスリッピングによるディフェクトマネージメントを行う場合、このフィールドにプライマリディフェクトリストが記録された位置に関する情報を格納し、スリッピングによるディフェクトマネージメントを行わない場合、全てのバイトに#00を設定する。ロケーションオブセカンダリディフェクトリスト(Location of Secondary Defect List:RBP 24)は、このパーティションでリニアリプレースメントによるディフェクトマネージメントを行う場合、このフィールドにセカンダリディフェクトリストが記録された位置に関する情報を格納し、リニ

アリプレースメントによるディフェクトマネージメントを行わない場合、全てのバイトに#00を設定する。リザーブド(Reserved:RBP 28)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。エクステンデッドデータアイデンティファイア(Extended Data Identifier:RBP 30)は、エクステンデッドデータフィールド、エクステンデッドデータエリアに記録されているエクステンデッドデータを特定するためのIDを指定する。エクステンデッドデータ(Extended Data:RBP 32)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0072】スベアエリアインフォメーション(Spare Area Information)は、表12に示す様式で記録する。

【0073】

【表12】

Spare Area Information

RBP	Length	Name	Contents
0	4	Start Physical Sector Number	Unit16
4	4	Number of Physical Sector	Unit16
8	8	Reserved	#00 bytes

【0074】スタートフィジカルセクタナンバ(Start Physical Sector Number:RBP 0)は、スベアエリアの先頭の物理セクタの物理セクタ番号を指定する。ナンバオブフィジカルセクタ(Number of Physical Sector:RBP 4)は、スベアを構成する物理セクタの数を指定する。リザーブド(Reserved:RBP 8)は、将来の拡張の為に予約さ

れ、全てのバイトに#00を設定する。

【0075】論理ボリュームインフォメーションヘッダ(Logical Volume Information Header)は、表13で示す様式で記録される。

【0076】

【表13】

Logical Volume Information Header

RBP	Length	Name	Contents
0	2	Character Set	Character Set
2	2	Logical Volume Name Size	Unit16
4	256	Logical Volume Name	bytes
260	2	Boot Indicator	Unit16
262	2	File System Indicator	Unit16
264	2	Logical Sector Size	Unit16
266	2	Number of Partitions (-Npv)	Unit16
268	4	Reserved	#00 bytes
272	16	Logical Volume Contents Use	bytes
288	14	Reserved	#00 bytes
302	2	Extended Data Identifier	Unit16
304	64	Extended Data	#00 bytes

【0077】キャラクタセット(Character Set:RBP 0)は、論理ボリューム名前フィールドに記録された論理ボリュームの名前の文字コードを指定する。ロジカルボリューム名前サイズ(Logical Volume Name Size:RBP 2)

は、論理ボリューム名前フィールドに指定された論理ボリュームの名前の大きさ(バイト数)を指定する。ロジカルボリューム名前(Logical Volume Name:RBP 4)は、論理ボリュームの名前を指定する。ブートインジケータ(Boo

t Indicator:RBP 260)は、論理ボリュームからの起動に関する情報を指定する。ブートインジケータの内容を表14に示す。ブートインジケータがアクティブであり、かつ、その先頭パーティションがその物理ボリュームにある

論理ボリュームは、物理ボリューム中に2つ以上あってはならない。

【0078】

【表14】

Boot Indicator

Value	Name	Contents
00h	Not Active	Physical volume is not set that computer boots up from this logical volume
80h	Active	Physical volume is set that computer boots up from this logical volume

【0079】ファイルシステムインジケータ(File System Indicator:RBP 262)は、この論理ボリュームで使用されているファイルシステムを指定する。ファイルシステ

ムインジケータの内容を表15に示す。

【0080】

【表15】

File System Indicator

Value	Name	Contents
00h	Unknown	This logical volume is unknown.
01h	12bit FAT	This logical volume is formatted with 12bit FAT.
04h	16bit FAT	This logical volume is formatted with 16bit FAT.
05h	16bit FAT,Extended	This logical volume is formatted with 16bit FAT, and defined an extended partition.
06h	16bit FAT,Extended, 64KB/claster	This logical volume is formatted with 16bit FAT, and defined an extended partition, using 64KB/claster.
07h	HPFS	This logical volume is formatted with HPFS.
08h	32bit FAT	This logical volume is formatted with 32bit FAT.
F0h	KIFS	This logical volume is formatted with KIFS.

【0081】ロジカルセクタサイズ(Logical Sector Size:RBP 264)は、この論理ボリュームの論理セクタの大きさ(バイト数)を指定する。ナンバオブパーティション(Number of Partitions:RBP 266)は、この論理ボリュームを構成するパーティションの数を指定し、パーティションマップの数と一致する。リザーブド(Reserved:RBP 268)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。ロジカルボリュームコンテンツユース(Logical Volume Contents Use:RBP 272)は、この論理ボリュームで使用されているファイルシステムが自由に使用してもよい領域である。リザーブド(Reserved (RBP 288)は将来の拡張の為に予約され、全てのバイトに#00を設定

する。エクステンデッドデータアイデンティファイア(Extended Data Identifier:RBP 302)は、エクステンデッドデータフィールド、エクステンデッドデータエリアに記録されているエクステンデッドデータを特定するためのIDを指定する。エクステンデッドデータ(Extended Data:RBP 304)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0082】パーティションマップ (Partition Map)は表16に示す様式で記録される。

【0083】

【表16】

Partition Map

RBP	Length	Name	Contents
0	20	Volume Identifier	bytes
20	2	Partition Number	Uint16
22	2	Reserved	#00 bytes

【0084】ボリュームアイデンティファイア (Volume Identifier:RBP 0)は、論理ボリュームを構成するパーティションが属している物理ボリュームの物理ボリュームインフォメーションに記録された物理ボリューム識別子を指定する。パーティションナンバ(Partition Number:RBP 2

0)は、論理ボリュームを構成するパーティションのパーティション番号を指定する。リザーブド(Reserved:RBP 22)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0085】ディフェクトリストインフォメーションへ

ッダ (Defect List Information Header) は表 17 に示す様式で記録される。

【0086】

【表 17】

Defect List Information Header

RBP	Length	Name	Contents
0	2	Number of MIB for Primary Defect List	Uint16
2	2	Number of MIB for Secondary Defect List	Uint16
4	12	Reserved	#00 bytes

【0087】ナンバオブMIBフォアプライマリディフェクトリスト (Number of MIB for Primary Defect List: RBP 0) は、プライマリディフェクトリストを記録するのに使用しているMIBの数を指定する。ナンバオブMIBフォアセコンダリディフェクトリスト (Number of MIB for Secondary Defect List: RBP 2) は、セコンダリディフェクトリストを記録するのに使用しているMIBの数を指定

する。リザーブド (Reserved: RBP 4) は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0088】プライマリディフェクトリスト / セコンダリディフェクトリスト (Primary Defect List/Secondary Defect List) は表 18 に示す様式で記録される。

【0089】

【表 18】

Primary Defect List / Secondary Defect List

RBP	Length	Name	Contents
0	8	Signature	Signature
8	2	Partition Number	Uint16
10	2	Number of Entries (=Npd)	Uint16
12	4	Reserved	#00 bytes
16	4(8)Npd	Defect List Entry	bytes

【0090】シグネチャ (Signature: BP 0) のデータタイプフィールドは、プライマリディフェクトリストの場合、18が設定され、セコンダリディフェクトリストの場合、19が設定される。パーティションナンバ (Partition Number: BP 8) は、このディフェクトリストを使用しているパーティションのパーティション番号を指定する。ナンバオブエントリーズ (Number of Entries: BP 10) は、ディフェクトリストエントリ (Defect List Entry) のエントリー数を指定する。リザーブド (Reserved: RBP 12) は、将来の拡張の為に予約され、全てのバイトに#00を設定する。ディフェクトリストエントリ (Defect List Entry: RBP 16) は、プライマリディフェクトリストの場合、プライマリディフェクトリストエントリを記録し、セコンダリディフェクトリストの場合、セコンダリ

ディフェクトリストエントリを記録する。ディフェクトリストエントリは、どちらの場合も、それぞれのエントリのフィジカルセクタナンバオブディフェクトセクタ (Physical Sector Number of Defect Sector) フィールドの値の昇順に記録する。

【0091】メディアインフォメーションディスクリプタ (Media Information Descriptor) の構造を図 9 に示す。

【0092】メディアインフォメーションディスクリプタヘッダ (Media Information Descriptor Header) は、表 19 に示す様式で記録される。

【0093】

【表 19】

Media information Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	Uint16
10	6	Reserved	#00 bytes
16	2	Number of discs	Uint16
18	2	Number of sides per disc	Uint16
20	2	Number of layers per side	Uint16
22	2	Number of zones per layer (=Nz)	Uint16
24	8	Reserved	#00 bytes
32	2	Number of cylinders	Uint16
34	2	Number of heads (tracks per cylinder)	Uint16
36	2	Number of sectors per tracks	Uint16
38	10	Reserved	#00 bytes

【0094】シグネチャ (Signature: BP 0) のデータタイプ

フィールドは、20が設定される。ディスクリプタサイ

ズ(Descriptor Size:BP 8)は、メディアインフォメーションディスクリプタの大きさ(MIB数)を指定する。リザーブド(Reserved:BP 10)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。ナンバオブディスクス(Number of discs:BP 16)は、ディスク数を指定する。ナンバオブサイダーズパーディスク(Number of sides per disc:BP 18)は、ディスクあたりのサイド数を指定する。ナンバオブレイヤパーサイド(Number of layers per side:BP 20)は、サイドあたりのレイヤ数を指定する。ナンバオブゾーンズパーレイヤ(Number of zones per layer:BP 22)は、レイヤあたりのゾーン数を指定する。リザーブド(Reserved:BP 24)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。ナンバオ

ブシリンダーズ(Number of cylinders:BP 32)は、シリンダ数を指定する。ナンバオブヘッズ(Number of heads (tracks per cylinder):BP 34)は、ヘッド数(シリンダあたりのトラック数)を指定する。ナンバオブセクタパートラック(Number of sectors per tracks:BP 36)は、トラックあたりのセクタ数を指定する。リザーブド(Reserved:BP 38)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0095】ゾーンインフォメーション (Zone Information) は、表20に示す様式で記録される。

【0096】

【表20】

Zone Information

RBP	Length	Name	Contents
0	4	Start Physical Sector Number	Uint16
4	4	Number of Physical Sector	Uint16
8	8	Reserved	#00 bytes

【0097】スタートフィジカルセクタナンバ(Start Physical Sector Number:RBP 0)は、ゾーンの先頭の物理セクタの物理セクタ番号を指定する。ナンバオブフィジカルセクタ(Number of Physical Sector:RBP 4)は、ゾーンを構成する物理セクタの数を指定する。リザーブド(Reserved:RBP 8)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0098】ドライブインフォメーションディスクリプ

タ (Drive Information Descriptor) の構造を図10に示す。

【0099】ドライブインフォメーションディスクリプタヘッダ (Drive Information Descriptor Header) は、表21に示す様式で記録される。

【0100】

【表21】

Drive Information Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	Uint16
10	1	Strategy Type	Uint8
11	5	Reserved	#00 bytes

【0101】シグネチャ(Signature:BP 0)のデータタイプフィールドは、21が設定される。ディスクリプタサイズ(Descriptor Size:BP 8)は、ドライブインフォメーションディスクリプタの大きさ(MIB数)を指定する。ストラテジタイプ(Strategy Type:BP 10)は、ストラテジタイプを指定する。リザーブド(Reserved:BP 11)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。

【0102】エクステンデッドデータディスクリプタ (Extended Data Descriptor) の構造を図11に示す。

ここで、@APSは、アライントゥフィジカルセクタ (Align to Physical Sector)を示し、そのデータは物理セクタにアライメントしなければならないことを示す。また、直前のデータの次のバイトからそのセクタの終りまでの領域は、#00が設定される。

【0103】エクステンデッドデータディスクリプタヘッダ (Extended Data Descriptor Header) は、表22に示す様式で記録される。

【0104】

【表22】

Extended Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	Unit16
10	6	Reserved	#00 bytes
16	2	Location of Extended Data for Physical Volume	Desc Extent
20	4Np	Location of Extended Data for Partitions	Desc Extent
20+4Np	4Nv	Location of Extended Data for Logical Volume	Desc Extent

【0105】シグネチャ(Signature:BP 0)のデータタイプフィールドは、22が設定される。ディスクリプタサイズ(Descriptor Size:BP 8)は、エクステンデッドデータディスクリプタの大きさ(MIB数)を指定する。リザーブド(Reserved:BP 10)は、将来の拡張の為に予約され、全てのバイトに#00を設定する。ロケーションオブエクステンデッドデータフォアフィジカルボリューム(Location of Extended Data for Physical Volume:BP 16)は、この物理ボリュームに関する拡張データが記録されている場所を指定する。ロケーションオブエクステンデッドデータフォアパーティション(Location of Extended Data for Partitions:BP 20)は、各パーティションに関する拡張データが記録されている場所を指定する。ロケーションオブエクステンデッドデータフォアロジカルボリューム(Location of Extended Data for Logical Volume:BP 20+4Np)は、各論理ボリュームに関する拡張データが記録されている場所を指定する。

【0106】つぎに、メディア交換のレベル(Levels of medium interchange)について述べる。メディア交換のレベル1は、以下の制限を設ける。すなわち、論理ボリュームは、同一の物理ボリュームに属するパーティションから構成される。同一の物理ボリュームに複数のパーティ

ションが定義される場合、パーティションの領域は、重なってはならない。論理ボリュームを構成するパーティションの物理セクタは、全て同じ物理セクタサイズを有する。論理セクタサイズは、物理セクタサイズの倍数であり、または、物理セクタサイズは、論理セクタサイズの倍数である。パーティションの大きさは、論理セクタサイズまたは物理セクタサイズの大きいほうの値の倍数である。ディフェクトマネジメントを行うパーティションは、必ず1つ以上のスベアエリアを確保する。リニアリプレースメントによるディフェクトマネジメントは、そのパーティション内に確保されたスベアエリアを代替データ領域として使用する。

【0107】メディア交換のレベル1は、制限がない。

【0108】つぎに、ボリュームストラクチャの例(Example of volume structure)について説明する。表23は、VDRの場合のFAT, ISO9660(with Joliet), ISO/IEC13346, KIFSのハイブリッドディスクのボリューム構造の例を示す表である。表23の◆は再配置不可能な位置固定情報であることを示す。

【0109】

【表23】

Example of volume structure (FAT, 9660, 13346, KIFS Hybrid)

PSN(hax)	Descriptor	Contents
0	[FAT] Partition Table	◆[FAT] Partition Table
-	-	-
c	[KIFS] Anchor Descriptor	◆[KIFS] Anchor
-	-	-
10	[9660] Primary Volume Descriptor	◆[9660/13346/KIFS] Volume recognition Sequence
11	[9660] Primary Volume Descriptor (Reserve)	
12	[9660] Supplementary Volume Descriptor (for Joliet)	
13	[9660] Volume Descriptor Set Terminator	
14	[13346] Beginning Extended Area Descriptor	
15	[13346] NSR Descriptor	
16	[13346] Terminating Extended Area Descriptor	
17	[13346] Beginning Extended Area Descriptor	
18	[KIFS] KIFS Descriptor	
19	[13346] Terminating Extended Area Descriptor	
-	-	-
30	[13346] Primary Volume Descriptor	[13346] Main Volume Descriptor Sequence Extent
31	[13346] Implementation Use Volume descriptor	
32	[13346] Partition Descriptor	
33	[13346] Logical Volume Descriptor	
34	[13346] Unallocated Space Descriptor	
35	[13346] Terminating Descriptor	
-	-	-
40	[13346] Primary Volume Descriptor	[13346] Reserve Volume Descriptor sequence Extent
41	[13346] Implementation Use Volume Descriptor	
42	[13346] Partition Descriptor	
43	[13346] Logical Volume Descriptor	
44	[13346] Unallocated Space Descriptor	
-	-	-
80	[KIFS] MIA Map for Main MIA	[KIFS] Main MIA
81	[KIFS] MIA Map for Reserved MIA	
82	[KIFS] Volume Structure Descriptor	
83	[KIFS] Primary Defect List	
84	[KIFS] Secondary Defect List	
85	[KIFS] Media Information Descriptor	
86	[KIFS] Drive Information Descriptor	
87	[KIFS] Extended Data Descriptor	
88	[KIFS] Extended Data	
-	-	-
c0	[KIFS] MIA Map for Reserve MIA	[KIFS] Reserve MIA
c1	[KIFS] MIA Map for Main MIA	
c2	[KIFS] Volume Structure Descriptor	
c3	[KIFS] Primary Defect List	
c4	[KIFS] Secondary Defect List	
c5	[KIFS] Defect Sector	
c6	[KIFS] Media Information Descriptor	
c7	[KIFS] Drive Information Descriptor	
c8	[KIFS] Extended Data Descriptor	
c9	[KIFS] Extended Data	
-	-	-
100	[13346] Anchor Volume Descriptor Pointer	◆[13346] Anchor
-	-	-
150	[KIFS] LOGICAL VOLUME	-
-	-	-
LPSN-19	-	-
-	-	-
LPSN-100	[13346] Anchor Volume Descriptor Pointer	◆[13346] Anchor
-	-	-
LPSN-20	[KIFS] Anchor Descriptor	◆[KIFS] Anchor
-	-	-
LPSN-c	[KIFS] Anchor Descriptor	◆[KIFS] Anchor
-	-	-
LPSN	[13346] Anchor Volume Descriptor Pointer	◆[13346] Anchor

【0110】次に論理ボリューム上に構成されるAVファイルシステム (AV File System) について説明する。論理セクタ番号 (Logical Sector Number) は、論理セクタを識別するためにつけられた番号である。論理ボリューム (Logical Volume) は、連続で昇順の0から始まる論理セクタ番号を持つ等しい大きさの論理セクタから構成された集合である。。

【0111】ファイルシステム管理用マネージメントインフォメーションエリア (Management Information Area (MIA)) は、AVファイルシステムの各種の制御情報を格納する論理ボリューム上の連続した複数の論理セクタ

らなる領域である。マネージメントインフォメーションブロック (Management Information Block (MIB)) は、MIA内の論理セクタである。マネージメントインフォメーションブロック番号 (Management Information Block Number (MIB番号)) は、マネージメントインフォメーションブロックの論理セクタ番号 NumberからそのMIAの先頭マネージメントインフォメーションブロックの論理セクタ番号を引いた値を有する。

【0112】つぎに、AVファイルシステムの全体について説明する。後述するAVファイルシステムディスクリプタ (AV File System Descriptor) は、1個の論理セクタ

内に記録され、論理ボリューム上のメインMIAとリザーブMIAの位置、大きさ、そして、メインMIAとリザーブMIA上のMIAマップの位置を指定する。AVファイルシステムディスクリプタの位置は、前述の論理ボリュームインフォメーションヘッダのロジカルボリュームコンテンツソース

(Logical Volume Contents Use:BP 284) フィールドに表24に示すように設定される。

【0113】

【表24】

Logical Volume Contents Use field

RBP	Length	Name	Contents
0	4	Main AV File System Descriptor Location	Uint32
4	4	Reserve AV File System Descriptor Location	Uint32
8	8	Reserved	#00 bytes

【0114】メインAVファイルシステムディスクリプタロケーション(Main AV File SystemDescriptor Location:RBP 0)は、AVファイルシステムディスクリプタの論理セクタ番号を指定する。リザーブAVファイルシステムディスクリプタロケーション(Reserve AV File System Descriptor Location:RBP 4)は、メインAVファイルシステムディスクリプタロケーションで指定されたのとは別の場所にあるAVファイルシステムディスクリプタの論理セクタ番号を指定する。もし、論理ボリューム上にAVファイルシステムディスクリプタが1個しか存在しない場合、リザーブAVファイルシステムディスクリプタロケーションには、#FFFFFFFがセットされる。リザーブド(Reserved:RBP 8)は、拡張のために予約されており、#00が設定される。

【0115】AVファイルシステムの各種の管理情報は、ファイルシステム管理用マネージメントインフォメーションエリア (Management Information Area:MIA)に記録される。信頼性確保の為、等しい内容の管理情報を持つMIAは、論理ボリューム上の2箇所に記録され、それぞれ、メインMIA、リザーブMIAと称する。メインMIAとリザーブMIAの位置、大きさ、MIA中のMIAマップの位置は、AVファイルシステムディスクリプタで規定される。MIA内の論理セクタは、マネージメントインフォメーションブロック (MIB)と称され、その論理セクタ番号のMIAの先頭MIBからのオフセットはマネージメントインフォメーションブロック番号 (MIB番号)と称される。

【0116】MIBの指定は、MIB番号が使用される。MIAは、欠陥などにより使用することの出来ないMIB、未使用のMIB、そしてデータ構造体である、MIAマップ(MIA Map)、ファイルテーブル(File Table)、アロケーションエクステンツテーブル(Allocation Extents Table、アロケーションストラテジテーブル(Allocation Strategy Table)、ディフェクトインフォメーションテーブル(Defect Information Table (Optional))、およびエクステンデッドアトリビュートテーブル(Extended Attribute Table (Optional))を格納するのに使われるMIBから構成される。MIA中のMIBがどの目的で使われているかはMIAマップに記録される。各種のデータ構造体は、ひと

つのMIB内、または複数のMIBに格納される。データ構造体が複数のMIBに記録される場合、どのMIBをどの順番で連結するかが、MIAマップ中のMap エントリフィールドに記録される。データ構造体がMIBの途中で終わった場合、データの終わりの次のバイトからそのMIBの最後のバイトまでは、#00が格納される。

【0117】AVファイルシステムにおいて、ファイルやディレクトリは、後述するファイルテーブルによって管理される。ファイルテーブルの構造は、ファイルテーブルヘッダ中のパラメータであるファイルテーブルストラクチャタイプ(File Table Structure Type)によって決定される。ファイルテーブルストラクチャタイプ0において、ファイルテーブルは、ファイルテーブルヘッダと1個以上のファイルレコードから構成される。ファイルレコードは、固定長のデータ領域で、ファイルレコードを識別するためのフィールド、ファイルレコードの種類を表すフィールド、作成、および修正日時を表すフィールド、データの位置と大きさを表すフィールド、属性を表すフィールド、ペアレントリンク (Parent Link) と称される親ファイルレコードを指すフィールド、ネクストリンク (Next Link) と称される兄弟ファイルレコードを指すフィールド、チャイルドリンク (Child Link) と称される子ファイルレコードを指すフィールド、並びにエクステンデッドアトリビュートレコードチェイン(Extended Attribute Record Chain)を指すフィールドから構成される。ファイルレコードは、ファイルレコード番号と称される番号が付され、ペアレントリンク、ネクストリンク、チャイルドリンクは、このファイルレコード番号を使って指定される。

【0118】ファイルテーブルストラクチャタイプ0では、ファイルテーブルの最初のファイルレコードがルートとなる図12に示されるような木構造が構築される。図中の円は、一つのファイルレコードを表しており、ルートのファイルレコードはルートファイルレコード (Root File Record) と称される。参照すべきデータを持たないファイルレコードは、ディレクトリと称され、データを持つファイルレコードはファイルと称される。ディレクトリばかりでなく、ファイルも子ファイルレコード

を有することが出来る。この階層構造は、図 13 に示されるようにチャイルドリンク (Child Link)、ネクストリンク (Next Link)、ペアレントリンク (Parent Link) を設定する事により実現される。

【0119】 ネックストリンクで構成されるファイルレコードのリストは、ファイルレコードチェーンと呼ばれ、このリスト中には同じファイルIDで、かつ同じファイルタイプを有するレコードが2つ以上あってはならない。サブファイルは、ファイルの一種で、親ファイルレコードの参照するデータの一部分をあたかも別のファイルであるかのように示す。アトリビュート (Attribute) フィールドのデータロケーションタイプ (Data Location Type) に10が設定されたファイルレコードは、サブファイルを表す。

【0120】 AVファイルシステムではアロケーションエクステント (Allocation Extent) という論理ボリューム上の連続した領域を単位としたデータの管理が実行される。アロケーションエクステントは、論理セクタの任意のバイトオフセットから始まりその論理セクタ内の任意のバイトオフセットで終了するか、あるいは引き続く0個以上の論理セクタを含み、それに続く論理セクタの任意のバイトオフセットで終了する。アロケーションエクステントの開始点、終了点、属性等はアロケーションエクステントテーブル中のアロケーションエクステントレコードに記録される。

【0121】 アロケーションエクステントテーブルには論理ボリューム上のすべてのアロケーションエクステントに対応するアロケーションエクステントレコードが、登録される。アロケーションエクステントレコードは、次のアロケーションエクステントレコードを指し示すフィールドを有し、このフィールドを使って複数のアロケーションエクステントレコードから成るリストが作成できる。このリストは、アロケーションエクステントレコードチェーンと称される。通常、ファイルデータは、アロケーションエクステントレコードチェーンに対応するアロケーションエクステントの順序つき集合として扱われる。

【0122】 アロケーションエクステントテーブル中の使用されていないアロケーションエクステントレコード (アロケーションエクステントレコードステータスが0のレコード) から作られたリストは、フリーアロケーションエクステントレコードチェーンと称され、アロケーションエクステントテーブルから簡単にたどる事ができる。また対応するアロケーションエクステント中に欠陥 (ディフェクト) セクタを含み再利用に問題があると判定されるアロケーションエクステントレコード (アロケーションエクステントレコードステータスに10を有するレコード) を集めて作成したリストをディフェクティブアロケーションエクステントレコードチェーンと称し、

このリストもアロケーションエクステントテーブルから簡単にたどる事ができる。

【0123】 アロケーションエクステントを論理ボリュームのどの位置に置くかはアロケーションストラテジ (Allocation Strategy) によって決定される。アロケーションストラテジテーブルは、複数のアロケーションストラテジを登録し、ファイル毎に異なるアロケーションストラテジを使用して、アロケーションエクステントを論理ボリューム上に配置する事ができる。各アロケーションストラテジが管理する領域の範囲、またはアロケーションストラテジが使用するパラメータは、アロケーションストラテジテーブル中のアロケーションストラテジレコードに記録される。ファイルテーブルストラクチャタイプ0において、アロケーションストラテジは、ファイルレコードごとに決定され、ファイルレコードのデータロケーションフィールドに記録される。このデータロケーションフィールドは、アロケーションエクステントの操作の際に参照され、対応するアロケーションストラテジが呼び出される。

【0124】 アロケーションストラテジタイプ0 (Allocation Strategy Type 0) およびアロケーションストラテジタイプ1 (Allocation Strategy Type 1) の2つのアロケーションストラテジタイプが、定義されている。アロケーションストラテジタイプ0は、インデックスデータなどの比較的小さなサイズのファイルを非連続的に取り扱う場合に適した方式であり、アロケーションストラテジタイプ1はMPEG等の連続的にデータの読み書きを行うのに適した方式である。

【0125】 ディフェクトインフォメーションテーブル (Defect Information Table) は、論理ボリューム内の欠陥セクタの論理セクタ番号を記録したテーブルであり、欠陥セクタの管理に使用できる。

【0126】 エクステンデッドアトリビュートテーブル (Extended Attribute Table) は、ファイルあるいはディレクトリの拡張属性をMIA中に保持するために使用できる。エクステンデッドアトリビュートテーブルは、エクステンデッドアトリビュートテーブルヘッダ、および1個以上のエクステンデッドアトリビュートテーブルレコードから構成される。エクステンデッドアトリビュートレコードは、リンクのためのフィールドを有する固定長のレコードで、複数のエクステンデッドアトリビュートレコードをリストとしたエクステンデッドアトリビュートレコードチェーンを作成できる。

【0127】 AVファイルシステムが使用するデータ構造の先頭は、シグネチャ (Signature) が設定される。シグネチャは表 25 に示すように記録される。

【0128】

【表 25】

Signature

RBP	Length	Name	Contents
0	4	Identification	bytes="AVFS"
4	1	Version	Uint8=1
5	1	Data type	Uint8
6	2	Reserved	#00 bytes

【0129】アイデンティフィケーション(Identification:RBP 0)は、文字列"AVFS"がISO/IEC 646に従って設定される。バージョン(Version:RBP 4)は、バージョン番号を指定し、1が設定される。データタイプ(Data type:RBP 5)は、データ構造体の種類を指定する。データ構造体の種類により、表26に示される値が設定される。

【0130】

【表26】

Data type

Value	Interpretation
0	Reserved
1	AV File System Descriptor
2	MIA Map
3	File Table
4	Allocation Extents Table
5	Allocation Strategy Table
6	Defect Information Table
7	Extended Attribute Table
8-255	Reserved

【0131】リザーブド(Reserved:RBP 6)は、拡張のために予約され、#00が設定される。シグネチャは、クラッシュリカバリのときデータ構造体を識別する為に使われる。

【0132】AVファイルシステムディスクリプタ (AV File System Descriptor) は、表27に示すように記録される。

【0133】

【表27】

AV File System Descriptor

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Location of Main MIA	Uint32
12	4	Location of Reserve MIA	Uint32
16	2	Length of Main MIA	Uint16
18	2	Length of Reserve MIA	Uint16
20	4	Creation Time	Time Stamp
24	4	Modification Time	Time Stamp
28	2	Number of MIA Map Sectors in Main MIA (=x ₁)	Uint16
30	2	Number of MIA Map Sectors in Reserve MIA (=x ₂)	Uint16
32	2x ₁	MIA Map Sectors in Main MIA	bytes
32+2x ₁	2x ₂	MIA Map Sectors in Reserve MIA	bytes

【0134】シグネチャのデータタイプフィールドは、1が設定される。ロケーションオブメインMIA(Location of Main MIA:BP 8)は、メインMIAの開始論理セクタ番号を指定する。ロケーションオブリザーブMIA(Location of Reserve MIA:BP 12)は、リザーブMIAの開始論理セクタ番号を指定する。レングスオブメインMIA(Length of Main MIA:BP 16)は、メインMIAのサイズを論理セクタ数で指定する。レングスオブリザーブMIA(Length of Reserve MIA:BP 18)は、リザーブMIAのサイズを論理セクタ数で指定する。クリエーションタイム(Creation Time:BP

20)は、AVファイルシステムディスクリプタを作成した日時を格納する。モディティフィケーションタイム(Modification Time:BP 24)は、AVファイルシステムディスクリプタを更新した日時を指定する。ナンバオブMIAマップセクタインメインMIA(Number of MIA Map Sectors in Main MIA:BP 28)は、メインMIAマップセクタズ(Main MIA Map Sectors:BP 32)に記述されたMIB番号の数を指定する。

【0135】ナンバオブMIAマップセクターズインリザーブMIA(Number of MIA Map Sectors in Reserve MIA:BP

30)は、リザーブMIAマップセクターズ(Reserve MIA Map Sectors:BP 32 +2x1)に記述されたMIB番号の数を指定する。MIAマップセクタインメインMIA(MIA Map Sectors in Main MIA:BP 32)は、メインMIA中のMIAマップを構成するMIBを指定し、MIAマップを構成するMIBのMIB番号が順に設定される。MIAマップセクターズインリザーブMIA(MIA Map Sectors in Reserve MIA:BP 32 +2x1)は、リザーブMIA中のMIAマップを構成するMIBを指定し、MIA

MIA Map

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Location of MIA Map	Uint16
10	2	Location of Allocation Strategy Table	Uint16
12	2	Location of File Table	Uint16
14	2	Location of Allocation Extents Table	Uint16
16	2	Location of Defect List Table	Uint16
18	2	Location of Extended Attribute Descriptor	Uint16
20	2	Reserved	bytes
22	2	Number of Map Entries (=x1)	Uint16
24	2x1	Map Entries	bytes

【0138】シグネチャ(Signature:BP 0)のデータタイプフィールドは、2が設定される。ロケーションオブMIAマップ(Location of MIA Map:BP 8)は、このMIA内にあるMIAマップの先頭MIBのMIB番号を指定する。ロケーションオブアロケーションストラテジテーブル(Location of Allocation Strategy Table:BP 10)は、このMIA内にあるアロケーションストラテジテーブルの先頭MIBのMIB番号を指定する。ロケーションファイルテーブル(Location of File Table:BP 12)は、このMIA内にあるファイルテーブルの先頭MIBのMIB番号を指定する。

【0139】ロケーションオブアロケーションエクステンステブル(Location of Allocation Extents Table:BP 14)は、このMIA内にあるアロケーションエクステンテブルの先頭MIBのMIB番号を指定する。ロケーションオブディフェクトリストテーブル(Location of Defect List Table:BP 16)は、このMIA内にあるディフェクトリストテーブルの先頭MIBのMIB番号を指定する。もしこのMIA内にディフェクトリストテーブルが存在しない場合、#FFFFがセットされる。ロケーションオブエクステンデッドアトリビュートディスクリプタ(Location of Extended Attribute Descriptor:BP 18)は、このMIA内にあるエクステンデッドアトリビュートディスクリプタの先頭MIBのMIB番号を指定する。もしこのMIA内にエクステンデッドアトリビュートディスクリプタが存在しない場合、#FFFFがセットされる。リザーブド(Reserved:BP 20)は、拡張のために予約されており、#00が設定される。

【0140】ナンバオブマップエントリーズ(Number of Map Entries:BP 22)は、(BP 24)からはじまるマップエントリのエントリ数を指定する。この数は、MIAに存在するMIBの数に等しく、#FFF0以下である。マップエント

マップを構成するMIBのMIB番号が順に設定される。

【0136】MIAマップ(MIA Map)は、MIA内のMIBの使用状況を示すのに使用される。MIAマップは、MIA内の各種のデータ構造体、欠陥などにより使用することの出来ないMIB、未使用MIBの位置を示す。MIAマップは、表28に示すように記録される。

【0137】

【表28】

リーズ(Map Entries:BP 24)は、このMIA内のMIBの使用状況を指定する。1つのマップエントリは、Uint16からなり、最初のマップエントリはMIAの最初のMIB、2番目のマップエントリは2番目のMIB...に対応する。

【0141】マップエントリの値は、表29に示す意味を有する。

【0142】

【表29】

Map entry value

Value	Interpretation
#0000-#FFEF	Next MIB Number
#FFF0	Unusable MIB
#FFF1	Unused MIB
#FFF2-#FFFE	Reserved
#FFFF	Last MIB of the data structure

【0143】もしデータ構造体が論理セクタサイズに等しいかあるいは小さく、1つのMIB内に格納される場合、そのMIBに対応するマップエントリに#FFFFが、セットされる。データ構造体が複数のMIBにわたって記録される場合、最後以外のMIBに対応するマップエントリには次のMIBのMIB番号が、最後のMIBに対応するマップエントリには#FFFFがセットされる。マップエントリの値が#FFF1であるMIBは、そのブロックが未使用である事を示し、データ構造体が新しいMIBを必要とする場合に使用する事が出来る。マップエントリの値が#FFF0であるMIBは、その使用に問題がある(欠陥セクタなど)ことを表す。

【0144】ファイルテーブル(File Table)は図14に示すようにファイルテーブルヘッダとファイルテーブルデータから構成される。ファイルテーブルデータの構造はファイルテーブルヘッダのFile Table Structure T

ypeフィールドによって決まる。

【0146】

【0145】ファイルテーブルヘッダ (File Table Header) は、表30に示すように記録される。

【表30】

File Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Length of File Table Data	Uint32
12	2	File Table Structure Type	Uint16
14	16	File Table Structure Type dependent information	bytes

【0147】シグネチャ (Signature:BP 0) のデータタイプフィールドは、3が設定される。レングスオブファイルテーブルデータ (Length of File Table Data:BP 8) は、ファイルテーブルデータの長さをバイト数で指定する。ファイルテーブルストラクチャタイプ (File Table Structure Type:BP 12) は、ファイルテーブルデータの構造を規定する。ファイルテーブルストラクチャタイプディペンデントインフォメーション (File Table Structure Type dependent information:BP 14) は、ファイルテーブルストラクチャタイプ毎に決められた情報が設定される。

【0148】ファイルテーブルストラクチャタイプ (File Table Structure Type) が0の場合、ファイルテーブルは図15に示すようにファイルテーブルヘッダと1個以上のファイルレコードから構成される。ファイルレ

コードは、0から始まる連続、昇順の番号が付され、この番号は、ファイルレコード番号と称される。ファイルレコードのリストは次のレコードのファイルレコード番号をネクストリンク (Next Link) フィールドに設定する事により作られ、このリストはファイルレコードチェーンと称される。ファイルテーブル内の使用されていないすべてのファイルレコードは、フリーファイルレコードチェーンと称されるファイルレコードチェーンを作成する。

【0149】ファイルテーブルストラクチャタイプが0の場合、ファイルテーブルヘッダ (File Table Header) は表31に示すように記録されなければならない。

【0150】

【表31】

File Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Length of File Table Data	Uint32
12	2	File Table Structure Type	Uint16 (=0)
14	2	Number of File Records	Uint16
16	2	First Free File Record	Uint16
18	14	Reserved	#00bytes

【0151】シグネチャ (Signature:BP 0) のデータタイプフィールドは、3が設定される。レングスオブファイルテーブルデータ (Length of File Table Data:BP 8) は、ファイルレコードの長さにナンバオブファイルレコードズ (Number of File Records:BP 14) をかけた数が設定される。ファイルテーブルストラクチャタイプ (File Table Structure Type:BP 12) は、0が設定される。ナンバオブファイルレコードズ (Number of File Records:BP 14) は、ファイルテーブルを構成するファイルレコード数を指定する。ファイルレコード数は、1以上#FFF0以下の

値をとる。ファーストフリーファイルレコードズ (First Free File Records:BP 14) は、フリーファイルレコードチェーンの最初の要素を指し、ファイルテーブル内にフリーなファイルレコードが存在しない場合、#FFFFが設定される。リザーブド (Reserved:BP 18) は、拡張のために予約されており、#00が設定される。

【0152】ファイルレコード (File Record) は、表32に示すように記録されなければならない。

【0153】

【表32】

File Record of File Structure Type 0

RBP	Length	Name	Contents
0	2	File ID	Uint16
2	2	File Type	Uint16
4	4	Attribute	Uint32
8	4	Creation Time	Time Stamp
12	4	Modification Time	Time Stamp
16	6	Data Length	Uint64
24	8	Data Location	bytes
32	2	Child Link	Uint16
34	2	Next Link	Uint16
36	2	Parent Link	Uint16
38	2	Extended Attribute Record Number	Uint16

【0154】ファイルID(File ID:RBP 0)は、ファイルレコードチェーン中の同じファイルタイプを持つファイルレコードを識別するための番号を指定する。ファイルタイプ(File Type:RBP 2)は、このファイルレコードの種類を指示するための番号を指定する。アトリビュート(Attribute:RBP 4)は、このファイルレコードまたはこのファイルレコードの参照するデータの属性を指定する。クリエーションタイム(Creation Time:RBP 8)は、このファイルレコードの作成日時を指定する。モディフィケーションタイム(Modification Time:RBP 12)は、このファイルレコードまたはファイルレコードの参照するデータの変更日時を指定する。データレングス(Data Length:RBP 16)は、データロケーション(Data Location:RBP 24)の参照するデータの長さをバイトで指定し、参照するデータがない場合には0をセットする。データロケーション(Data Location:RBP 24)は、このファイルレコードの参照するデータの位置を指定する。フィールドの解釈は、アトリビュート(Attribute:RBP 4)のデータロケーションタイプ(Data Location Type:Bit 1-2)の内容によって変化する。チャイルドリンク(Child Link:RBP 32)は、チャイルドファイルレコードのファイルレコード番号を指定し、そのようなファイルレコードが存在しない場合、#FFFFが設定される。ネクストリンク(Next Link:RBP 34)は、ファイルレコードチェーンを構成する次のファイルレコードのファイルレコード番号を指定し、このファイルレコードがファイルレコードチェーンの最後の要素の場合、#FFFFが設定される。

【0155】ペアレントリンク(Parent Link:RBP 36)は、ペアレントファイルレコードのファイルレコード番号を指定し、このファイルレコードがルートファイルレコードである場合、自分自身のファイルレコード番号すなわち0が設定される。エクステンデッドアトリビュートレコードナンバ(Extended Attribute Record Number:RBP 38)は、このファイルレコードの使うエクステンデッドアトリビュートレコードチェーンの先頭のエクステンデッドアトリビュートレコード番号を指定し、エクステンデッドアトリビュートレコードを参照しない場合、#FFFFが設定される。

【0156】アトリビュート (Attribute) フィールドは、表33に示すように記録される。

【0157】

【表33】

Attribute of the File Record

Bit	Interpretation
0	Valid
1-2	Data Location Type
3	Protected
4	Sorted
5-31	Reserved

【0158】バリッド(Valid:Bit 0)は、このファイルレコードが有効なレコードであるかどうかを表し、0の場合、このファイルレコードが使われていないことを表し、ファイルレコードは、フリーファイルレコードチェーン中にある。バリッドが、1の場合、このファイルレコードが使用されていることを表し、ルートファイルレコードからチャイルドリンク、ネクストリンクを経て到達する事ができる。データロケーションタイプ(Data Location Type:Bit 0-1)は、データロケーション(Data Location:RBP 24)のフォーマットを指定する。データロケーションタイプが00の場合、データロケーションは参照するものがない事を示す(ファイルレコードがディレクトリの場合は、この値をセットする)。データロケーションタイプが01の場合、データロケーションは、アロケーションエクステンツレコードチェーンの先頭のアロケーションエクステンツレコード番号とアロケーションストラテジ番号を表34に示すフォーマットで表される。データロケーションタイプが10の場合、ファイルレコードはサブファイルであることを表し、データロケーションは、ペアレントファイルレコードのデータロケーションが表すデータの先頭からのオフセットがUint64で表される。11のデータロケーションタイプは、拡張のために予約されている。

【0159】

【表34】

Data Location file of Type 01

RBP	Length	Name	Contents
0	2	Reserved	#00 bytes
2	2	Allocation Strategy Number	Uint16
4	4	First Allocation Extent Record Number	Uint32

【0160】プロテクトド(Protected:Bit 3)は、このファイルレコードがプロテクトされていることを表す。ソーテッド(Sorted:Bit 4)は、このファイルレコードの属するファイルレコードチェーンがファイルタイプの若い順にソートされ、さらに同じファイルタイプの中ではファイルIDの若い順にソートされている事を表す。リザーブド(Reserved:Bit 5-31)は、拡張のために予約されている。

【0161】アロケーションエクステントテーブル (Structure of the Allocation Extents Table) は、図16に示すようにアロケーションエクステントテーブルヘッダとアロケーションエクステントレコードから構成される。アロケーションエクステントレコードには0から

始まる連続、昇順の番号が付される。この番号は、アロケーションエクステントレコード番号と称される。次のレコードのアロケーションエクステントレコード番号をネクストアロケーションエクステントレコードフィールドに設定する事により、アロケーションエクステントレコードのリストはつくられる。このリストはアロケーションエクステントレコードチェーンと称される。

【0162】アロケーションエクステントテーブルヘッダ (Allocation Extents Table Header) は、表35に示すように記録される。

【0163】

【表35】

Allocation Extents Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Number of Allocation Extent Records	Uint32
12	4	First Free Allocation Extent Record	Uint32
16	4	First Defective Allocation Extent	Uint32
20	4	Reserved	#00 bytes

【0164】シグネチャ(Signature:BP 0)のデータタイプフィールドは、4が設定される。ナンバオブアロケーションエクステントレコードズ(Number of Allocation ExtentRecords (BP 8)はアロケーションエクステントテーブル中のアロケーションエクステントレコードの数を指定する。ファーストフリーアロケーションエクステントレコード(First Free Allocation Extent Record:BP 12)は、フリーアロケーションエクステントレコードチェーンの最初の要素を指す。

【0165】アロケーションエクステントテーブル内にフリーなアロケーションエクステントレコードが存在しない場合、#FFFFFFFがこのフィールドに設定される。ファーストディフェクティブアロケーションエクステントレコード(First DefectiveAllocation Extent Record:BP 16)は、ディフェクティブアロケーションエクステ

ントレコードチェーンの最初の要素を指す。アロケーションエクステントテーブル内にディフェクティブアロケーションエクステントレコードが存在しない場合、#FFFFFFFがこのフィールドに設定される。リザーブド(Reserved:BP 20)は、拡張のために予約され、#00が設定される。

【0166】アロケーションエクステントレコード (Allocation Extent Record) は、アロケーションエクステントの開始位置、終了位置、属性、アロケーションエクステントレコードチェーンを構成する次のアロケーションエクステントレコードの位置を表す。アロケーションエクステントレコードは、表36に示すように記録される。

【0167】

【表36】

Allocation Extent Record

RBP	Length	Name	Contents
0	4	Start Logical Sector Number	Uint32
4	1	Allocation Strategy Number	Uint8
5	1	Reserved	Uint8
6	2	Start Offset	Uint16
8	4	End Logical Sector Number	Uint32
12	2	Reserved	Uint16
14	2	End Offset	Uint16
16	4	Attribute	Uint32
20	4	Next Allocation Extent Record	Uint32
24	8	Length of the Allocation Extent	Uint64

【0168】スタートロジカルセクタナンバ(Start Logical Sector Number:RBP 0)は、アロケーションエクステン트의開始バイトを含む論理セクタを指定し、論理セクタ番号が設定される。アロケーションストラテジナンバ(Allocation Strategy Number:RBP 4)は、このアロケーションエクステントレコードがどのアロケーションストラテジに従って配置されているかを指示する。リザーブド(Reserved:RBP 5)は、拡張のために予約され、#00が設定される。スタートオフセット(StartOffset:RBP 6)は、アロケーションエクステン트의開始バイトを含む論理セクタの先頭バイトから開始バイトまでのバイトオフセットを指定し、開始位置がその論理セクタの先頭バイトに等しければ0がセットされる。

【0169】エンドロジカルセクタナンバ(End Logical Sector Number:RBP 8)は、アロケーションエクステン트의最終バイトを含む論理セクタの論理セクタ番号を指定する。リザーブド(Reserved:RBP 12)は、拡張のために予約され、#00が設定される。エンドオフセット(End Offset:RBP 14)は、アロケーションエクステン트의終了バイトを含む論理セクタの先頭バイトから終了バイトまでのオフセットを指定し、終了バイトがその論理セクタの先頭バイトに等しいならば0がセットされる。アトリビュート(Attribute:RBP 16)の表す値は、表37に示す意味を有する。

【0170】

【表37】

Attribute of the Allocation Extent Record

Bit	Interpretation
0-1	Allocation Extnt Record Status
1-31	Reserved

【0171】アロケーションエクステントレコードステータス(Bit 0-1)が01の場合、このアロケーションエクステントレコードは、有効なアロケーションエクステントを指し、正常に読み出しができる。このビットが11の場合、このアロケーションエクステントレコードは有効なアロケーションエクステントを指しており、かつ欠陥セクタの存在などにより、正常に読み出しを出来ない可

能性のある事を表す。このビットが00の場合、このアロケーションエクステントレコードは現在使用されておらず、新しいアロケーションエクステントを配置する際に使用出来る事を表す。このビットが10の場合、このアロケーションエクステントレコードの指すアロケーションエクステントは、どこからも参照されていないが、欠陥セクタを含んでいるために新しいアロケーションエクステントを配置する為に使用するのとは適当でない事を表す。リザーブド(Reserved:Bit 2-31)は、拡張のために予約されており、0が設定される。

【0172】ネクストアロケーションエクステントレコード(Next Allocation Extent Record:RBP 20)は、アロケーションエクステントレコードチェーンを構成する次のアロケーションエクステントレコード番号を指定する。アロケーションエクステントレコードがアロケーションエクステントレコードチェーンの最後の要素である場合、#FFFFFFFがセットされる。レングスオブザアロケーションエクステント(Length of the Allocation Extent:RBP 24)は、このアロケーションエクステントレコードが指示するアロケーションエクステントの長さをバイト数で指示する。スタートロジカルセクタナンバ(Start Logical Sector Number:RBP 0)、スタートオフセット(Start Offset:RBP 6)、エンドロジカルセクタナンバ(End Logical Sector Number:RBP 8)、およびエンドオフセット(End Offset:RBP 14)から計算で求められるバイト数とこのフィールドにセットされたバイト数は等しい。

【0173】アロケーションストラテジテーブルはAVファイルシステムがこの論理ボリュームでデータを配置するのに使用しているすべてのアロケーションストラテジを指定する。アロケーションストラテジテーブルは図17に示すようにアロケーションストラテジテーブルヘッダとアロケーションストラテジレコードから構成される。

【0174】アロケーションストラテジテーブルヘッダ(Allocation Strategy Table Header)は、表38に示すように記録される。

【0175】

【表38】

Allocation Strategy Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Number of Allocation Strategy Record	Uint16
10	6	Reserved	#00 bytes

【0176】シグネチャ(Signature:BP 0)のデータタイプフィールドは、5が設定される。ナンバオブアロケーションストラテジレコード(Number of Allocation Strategy Record:BP 8)は、アロケーションストラテジテーブル中のアロケーションストラテジレコードの数を指定する。リザーブド(Reserved:RBP 10)は、拡張のために予約され、#00が設定される。

【0177】アロケーションストラテジレコードは、アロケーションストラテジを指定するのに使用される。アロケーションストラテジレコードは、表39に示すように記録される。

【0178】

【表39】

Allocation Strategy Record

RBP	Length	Name	Contents
0	2	Length of Allocation Strategy Record	Uint16
2	2	Allocation Strategy Type	Uint16
4	1	Allocation Strategy Number	Uint8
5	3	Reserved	#00 bytes
8	x1	Allocation Strategy Type Dependent Data	bytes

【0179】レングスオブアロケーションストラテジレコード(Length of Allocation Strategy Record:RBP 0)は、このアロケーションストラテジレコードの長さをバイト数で指定し、その長さは8の倍数である。アロケーションストラテジタイプ(Allocation Strategy Type:RBP 2)は、このアロケーションストラテジレコードの種類を指定する。アロケーションストラテジナンバ(Allocation Strategy Number:RBP 4)は、このアロケーションストラテジレコードがアロケーションストラテジテーブル中の何番目のレコードであるかを指定し、このレコードが最初のレコードならば0がセットされる。リザーブド(Reserved:RBP 5)は、拡張のために予約されており、#00が設定されなければならない。アロケーションストラテジタイプディペンデントデータ(Allocation Strategy Type Dependent Data:RBP 8)は、アロケーションストラテジタイプ毎に決まった内容がセットされる。

【0180】アロケーションストラテジタイプ0においては、次の条件を満足する。第1に、アロケーションエクステントは、アロケーションストラテジレコードのスタートロジカルセクタナンバ(Start Logical Sector Number:RBP 8)およびエンドロジカルセクタナンバ(End Logical Sector Number:RBP 12)で指定された領域内に配置されなければならない。第2に、論理セクタの一部が、あるアロケーションエクステントに割り当てられている場合、その論理セクタのどのバイトも別のアロケーションエクステントに属さない。第3に、アロケーションエクステントの先頭と論理セクタの先頭は一致する。アロケーションストラテジタイプ0のアロケーションストラテジレコードは、表40に示すように記録される。

【0181】

【表40】

Allocation Strategy Record of Allocation Strategy Type 0

RBP	Length	Name	Contents
0	2	Length of Allocation Strategy Record	Uint16(=16)
2	2	Allocation Strategy Type	Uint16(=0)
4	1	Allocation Strategy Number	Uint8
5	3	Reserved	#00 bytes
8	4	Start Logical Sector Number	Uint32
12	4	End Logical Sector Number	Uint32

【0182】レングスオブアロケーションストラテジレコード(Length of Allocation Strategy Record:RBP 0)は、16が設定される。アロケーションストラテジタイプ(Allocation Strategy Type:RBP 2)は、0が設定さ

れる。アロケーションストラテジナンバ(Allocation Strategy Number:RBP 4)は、このアロケーションストラテジレコードがアロケーションストラテジテーブル中の何番目のレコードであるかを指定し、このレコード

が最初のレコードならば0がセットされる。リザーブド(Reserved:RBP 5)は、拡張のために予約され、#00が設定される。スタートロジカルセクタナンバ(Start Logical Sector Number:RBP 8)は、アロケーションエクステントを配置する領域の先頭論理セクタ番号を指定する。エンドロジカルセクタナンバ(End Logical Sector Number:RBP 12)は、アロケーションエクステントを配置する

領域の最後の論理セクタ番号を指定する。

【0183】アロケーションストラテジタイプ1のアロケーションストラテジレコードは、表41に示すように記録される。

【0184】

【表41】

Allocation Strategy Record of Allocation Strategy Type 1

RBP	Length	Name	Contents
0	2	Length of Allocation Strategy Record	Uint16
2	2	Allocation Strategy Type	Uint16 (=1)
4	1	Allocation Strategy Number	Uint8
5	3	Reserved	#00 bytes
8	2	Number of Zones (=x ₁)	Uint16
10	6	Reserved	#00 bytes
16	16x ₁	Zone Information Records	

【0185】レングスオブアロケーションストラテジレコード(Length of Allocation Strategy Record:RBP 0)は、このアロケーションストラテジレコードの長さ、16+16x₁が設定される。アロケーションストラテジタイプ(Allocation Strategy Type:RBP 2)は、1が設定される。アロケーションストラテジナンバ(Allocation Strategy Number:RBP 4)は、このアロケーションストラテジレコードがアロケーションストラテジテーブル中の何番目のレコードであるかを指定し、このレコードが最初のレコードならば0がセットされる。リザーブド(Reserved:RBP 5)は、拡張のために予約され、#00が

設定される。ナンバオブゾーン(Number of Zones:RBP 8)は、アロケーションストラテジレコード中のゾーンインフォメーションレコードの数を指定する。リザーブド(Reserved:RBP 10)は、拡張のために予約され、#00が設定される。ゾーンインフォメーションレコードズ(Zone Information Records:BP 16)は、ナンバオブゾーン(Number of Zones:RBP 8)で指定された数のゾーンインフォメーションレコードが設定される。ゾーンインフォメーションレコードは、表42に示すように記録される。

【0186】

【表42】

Zone Information Record

RBP	Length	Name	Contents
0	4	Start Logical Sector Number	Uint32
4	4	End Logical Sector Number	Uint32
8	4	Length of Allocation Unit	Uint32
12	4	Reserved	#00 bytes

【0187】スタートロジカルセクタナンバ(Start Logical Sector Number:RBP 0)は、このゾーンの開始論理セクタ番号を指定する。エンドロジカルセクタナンバ(End Logical Sector Number:RBP 4)は、このゾーンの最終論理セクタ番号を指定する。レングスオブアロケーションユニット(Length of Allocation Unit:RBP 8)は、このゾーン内に配置を行う際のアロケーションユニットを指定する。リザーブド(Reserved:RBP 12)は、拡張の

ために予約され、#00が設定される。

【0188】ディフェクトインフォメーションテーブル(Defect Information Table)は論理ボリューム中の欠陥セクタの論理セクタ番号を記録する。ディフェクトインフォメーションテーブルは、表43に示すように記録される。

【0189】

【表43】

Defect Information Table

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Number of Defect Sectors (=x ₁)	Uint32
12	4	Reserved	#00 bytes
16	4x ₁	Defect Sector Addresses	bytes

【0190】シグネチャ(Signature:BP 0)のデータタイプフィールドは、6が設定される。ナンバオブディフェクトセクタズ(Number of Defect Sectors:BP 8)は、(BP

16)からはじまるディフェクトセクタアドレスのエントリ数を指定する。リザーブド(Reserved:BP 12)は、拡張のために予約され、#00が設定される。ディフェクトセ

クタアドレス(Defect Sector Addresses:BP 16)は、この論理ボリューム中で検出されたディフェクトセクタの論理セクタ番号を指定し、1つのエントリはUint32からり、ここに記録される値は昇順にソートされている。

【0191】エクステンデッドアトリビュートテーブル(Extended Attribute Table)は、図18に示すようにエクステンデッドアトリビュートテーブルヘッダとエクステンデッドアトリビュートレコードから構成される。エクステンデッドアトリビュートテーブル中のエクステンデッドアトリビュートレコードは、0から始まる連続、昇順の番号が付され、この番号は、エクステンデッドアトリビュートレコード番号と称される。エクステンデッドアトリビュートレコードのリストは、ネクストエ

クステンデッドアトリビュートレコードフィールドに次のレコードを設定する事により作成され、このリストは、エクステンデッドアトリビュートレコードチェーンと称される。エクステンデッドアトリビュートテーブル内の使用されてないエクステンデッドアトリビュートレコードは、フリーエクステンデッドアトリビュートレコードチェーンと呼ばれるリストを作成する。

【0192】エクステンデッドアトリビュートテーブルヘッダ(Extended Attribute TableHeader)は、表44に示すように記録される。

【0193】

【表44】

Extended Attribute Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Number of Extended Attribute Record	Uint16
10	2	First Free Extended Attribute Record	Uint16
12	4	Reserved	#00 bytes

【0194】シグネチャ(Signature:BP 0)のデータタイプフィールドは、7が設定される。ナンバオブエクステンデッドアトリビュートレコード(Number of Extended Attribute Record:BP 8)は、エクステンデッドアトリビュートテーブル中のエクステンデッドアトリビュートレコードの数を指定し、#FFF0以下である。ファーストフリーエクステンデッドアトリビュートレコード(First Free Extended Attribute Record:BP 10)は、フリーエクステンデッドアトリビュートレコードチェーンの最初の要素を指し、エクステンデッドアトリビュートテーブル

内にフリーのエクステンデッドアトリビュートレコードが存在しない場合、#FFFFが設定される。リザーブド(Reserved:BP 12)は、拡張のために予約され、#00が設定される。

【0195】エクステンデッドアトリビュートレコード(Extended Attribute Record)は、表45に示すように記録される。

【0196】

【表45】

Extended Attribute Record

RBP	Length	Name	Contents
0	2	Next Extended Attribute Record	Uint16
2	30	Extended Attribute Information	bytes

【0197】ネクストエクステンデッドアトリビュートレコード(Next Extended AttributeRecord:RBP 0)は、エクステンデッドアトリビュートレコードチェーンを構成する次のエクステンデッドアトリビュートレコード番号を指定し、このエクステンデッドアトリビュートレコードが最後のエクステンデッドアトリビュートレコードである場合、#FFFFがセットされる。

【0198】既存のファイルシステムの多くはメディアの欠陥セクタ処理をファイルシステムの下に位置するレイヤ(例えばドライブ内部の交替処理)で行う事を前提に設計されている。これらのファイルシステムでは欠陥セクタがどこにあるのかが分からず、欠陥がない部分ではドライブの生の転送速度でデータにアクセスできるが、交替処理が行われている部分ではそれよりもはるかに低い転送速度でしかアクセスが出来ない。

【0199】従来のコンピュータ用途では平均アクセス

時間の向上が要求される事はあっても個々のアクセス時間の見積もりが要求される事はなかったため上記のような構成でも問題はなかった。しかし、オーディオ、ビデオ用途ではデータを一定時間内に一定量供給できなければ音声や映像を正しく記録再生する事ができないため、ファイルシステムがデータアクセスにかかる時間の見積もりを行える事が必要となってきた。

【0200】そこで本ファイルシステムでは欠陥セクタ処理を下のレイヤで行わなくても良いという前提を導入し、ファイルシステムがデータのアクセスにかかる時間を正確に見積もる事が出来るようにした。これに伴い、本ファイルシステムでは従来のファイルシステムにはなかった欠陥セクタ処理の為にフィールドやフラグが用意され、これを使って欠陥セクタの処理が行う事が出来る。ここでは本ファイルシステムに用意された機能を使って欠陥セクタ処理を行う方法の一例を解説する。

【0201】一般に欠陥セクタが検出されるのは次のいずれかである。第1に、書き込み中にエラーが発生し欠陥セクタが検出される。第2に、書き込みは正常終了したが、書き込み直後にその部分を読み出した際にエラーが検出される。第3に、書き込み、および書き込み直後の読み出しは正常終了したが、時間を経て読み出しを行った時にエラーが検出される。

【0202】第1および第2の場合は、書き込み直後に読み出しを行い、正しく書き込めた事を確認する(Write and Verify)操作を行うことにより書き込み時に検出、対応できる。

【0203】第3の場合は、光ディスクにおけるごみ、傷による障害などで発生するケースである。このケースについては完全な対応策は存在しないが、多重書きを行う事により、データロスの可能性を著しく下げる事が出来る。本ファイルシステムは、主にこのWrite and Verifyと多重書きの2つの手法を使って欠陥セクタの処理を行う。

【0204】ポリウム構造は、ポリウムストラクチャディスクリプタ(Volume Structure Descriptor)、メディアインフォメーションディスクリプタ(Media Information Descriptor)、ドライブインフォメーションディスクリプタ(Drive Information Descriptor)、およびエクステンデッドデータディスクリプタ(Extended Data Descriptor)により定義される。これらの情報に対する欠陥セクタへの対応は、以下のように行う。

【0205】ポリウムストラクチャディスクリプタ、メディアインフォメーションディスクリプタ、ドライブインフォメーションディスクリプタ、およびエクステンデッドデータディスクリプタは、MIAにより管理される。MIAは、記録の際に必ずwriteand verifyを行うことで確実に非欠陥セクタに記録することが可能である。また、MIAは、記録後に生じた欠陥を考慮し、MIAを2箇所に重複記録し、MIA内の使用状況を管理するMIAマップについても2箇所に重複記録する。

【0206】さらに、ポリウム管理システムによって定義される論理ポリウムでは、これを構成するパーティション毎にスリッピング、リニアリプレースメントによるディフェクトマネージメントが行える。

【0207】AVファイルシステムの、欠陥セクタへの対応は以下のように行う。AVファイルシステムは、AVファイルシステムディスクリプタに書き込みを行うとき、Writeand Verifyを実行し、正しく書き込めた事を確認し、書き込みに失敗した場合、別の場所にAVファイルシステムディスクリプタを書き、ロジカルポリウムコンテンツユースフィールドの内容を書き換える。また、AVファイルシステムディスクリプタを、2箇所に書くことにより信頼性を向上させる。

【0208】AVファイルシステムは、MIA内のセクタの書き込みを行うとき、Write and Verifyを実行し、正し

く書き込めた事を確認し、書き込みに失敗した場合、MIAマップのエントリフィールドに#FFF0を書き込み、別のMIA内のセクタに対して同じシーケンスを実行する。また、AVファイルシステムは、MIA自体を論理ポリウム上の2箇所に書くことにより信頼性を向上させる。

【0209】AVファイルシステムが動作中に検出した欠陥セクタは、ディフェクトインフォメーションテーブルに登録され、次回からそのセクタは、使用しないようにする事が出来る。

【0210】アロケーションエクステントに登録されるデータは、転送速度の要求からWrite and Verify(ライトアンドベリファイ)オペレーションが行えず、Write(ライト)オペレーションのみを実行するときがある。いずれの場合も欠陥セクタを検出した場合、AVファイルシステムは、その部分を独立したアロケーションエクステントとして、そのアロケーションエクステントレコードのアロケーションエクステントレコードステータスに10が設定され、そのアロケーションエクステントを、ディフェクティブアロケーションエクステントレコードチェーンに入れる。読み出し時にアロケーションエクステント中に欠陥セクタを検出した場合、AVファイルシステムは、アロケーションエクステントレコードステータスに11をセットする。このアロケーションエクステントの解放が行われるとき、欠陥セクタが調べられ、その欠陥セクタの部分は、アロケーションエクステントレコードステータスが10のアロケーションエクステントとして、ディフェクティブアロケーションエクステントレコードチェーンに登録される。

【0211】図19は、本発明の記録再生装置1の一実施の形態の構成を示すブロック図である。記録再生装置1は、光ディスク8が装着され、光ディスク8に外部から供給されたビデオ信号およびオーディオ信号並びにPC(Personal Computer)データを記録するか、または、光ディスク8に記録されている信号を読み取り、外部に出力する。

【0212】ユーザ入出力部2は、キーパネル11およびLCD(Liquid Crystal Display)12を有する。キーパネル11は、ユーザの操作に応じた信号を発生し、システムコントロール部5に供給するようになされている。LCD12は、システムコントロール部5から供給された信号に基づき、記録再生装置1の状態または記録再生装置1に装着された光ディスク8に関する情報等を表示する。

【0213】AV入出力部3は、エンコーダ/デコーダ13および14並びにマルチプレクサ/デマルチプレクサ15を有し、システムコントロール部5から供給された信号に基づき、エンコーダ/デコーダ13および14並びにマルチプレクサ/デマルチプレクサ15を制御する。また、AV入出力部3は、システムコントロール部5にエンコーダ/デコーダ13および14並びにマルチプ

レクサ/デマルチプレクサ 15 の状態を示す信号を供給する。

【0214】エンコーダ/デコーダ 13 は、記録時において、外部から供給されたビデオ信号を圧縮（エンコード）して、ビデオ信号に対応する所定の方式のビデオデータをマルチプレクサ/デマルチプレクサ 15 に出力し、再生時において、マルチプレクサ/デマルチプレクサ 15 から供給された所定の方式のビデオデータを伸張（デコード）して外部に出力する。エンコーダ/デコーダ 14 は、記録時において、外部から供給されたオーディオ信号を圧縮（エンコード）して、オーディオ信号に対応する所定の方式のオーディオデータをマルチプレクサ/デマルチプレクサ 15 に出力し、再生時において、マルチプレクサ/デマルチプレクサ 15 から供給された所定の方式のオーディオデータを伸張（デコード）して外部に出力する。

【0215】マルチプレクサ/デマルチプレクサ 15 は、記録時において、エンコーダ/デコーダ 13 および 14 から供給された所定の方式のビデオデータおよびオーディオデータを多重化し、ドライブ部 7 に出力するようになされている。また、再生時において、ドライブ部 7 から供給された多重化されたビデオデータおよびオーディオデータを分離し、ビデオデータをエンコーダ/デコーダ 13 に、オーディオデータをエンコーダ/デコーダ 14 に出力するようになされている。

【0216】PCデータ入出力部 4 は、インターフェース 16 を有し、システムコントロール部 5 から供給された信号に基づき、インターフェース 16 を制御し、システムコントロール部 5 にインターフェース 16 の状態を示す信号を出力する。インターフェース 16 は、外部のパーソナルコンピュータ（図示せず）等から供給された所定の形式のPCデータを入力し、ドライブ部 7 が読み取り可能なデータに変更し、ドライブ部 7 に出力する。インターフェース 16 は、また、ドライブ部 7 から供給されたデータを所定の形式で、外部のパーソナルコンピュータ等に出力するようになされている。

【0217】システムコントロール部 5 は、ユーザ入出力部 2、AV入出力部 3、PCデータ入出力部 4、およびファイル管理部 6 それぞれの状態に基づき、ユーザ入出力部 2、AV入出力部 3、PCデータ入出力部 4、およびファイル管理部 6 を制御するようになされている。

【0218】ファイル管理部 6 は、システムコントロール部 5 からの信号に基づき、ドライブ部 7 を制御し、ドライブ部 7 の状態に応じた信号をシステムコントロール部 5 に供給するようになされている。

【0219】ドライブ部 7 は、バッファ 17、ECC回路 18、変調/復調回路 19、およびピックアップ 20 を有し、ファイル管理部 6 からの信号に基づき、バッファ 17、ECC回路 18、変調/復調回路 19、およびピックアップ 20 を動作させ、光ディスク 8 に信号を記録し、

または光ディスク 8 から信号を読み出すようになされている。

【0220】バッファ 17 は、AV入出力部 3 またはPCデータ入出力部 4 から供給されたデータを一時的に記憶し、データが途切れないように、ECC(Error Correction Code)回路 18 にデータを出力し、また、ECC回路 18 から供給されたデータを一時的に記憶し、データが途切れないように、AV入出力部 3 またはPCデータ入出力部 4 に供給するようになされている。

【0221】ECC回路 18 は、バッファ 17 から供給されたデータにECCを付加して、変調/復調回路 19 に出力し、また、変調/復調回路 19 から供給されたデータを、ECCを基に誤り訂正した後、バッファ 17 に出力するようになされている。

【0222】変調/復調回路 19 は、ECC回路 18 から供給されたデータを所定の方式に変調してピックアップ 20 に出力し、ピックアップ 20 から供給されたデータを所定の方式に基づいて復調し、ECC回路 18 に出力するようになされている。

【0223】ピックアップ 20 は、変調/復調回路 19 から供給されたデータに基づき、記録再生装置 1 に装着された光ディスク 8 にデータを記録し、または光ディスク 8 に記録されたデータを読み取り、変調/復調回路 19 に出力するようになされている。

【0224】図 20 は、再生のときの、バッファ 17 に記録されているデータの量とバッファ 17 に書き込まれるデータの速度の関係を示す図である。バッファ 17 から出力されるデータの読み出し速度 R_{out} は、エンコーダ/デコーダ 13 および 14 が信号の出力を途切れさせないようにするため、所定の値以上の一定値となるように制御される。バッファ 17 に供給されるデータのデータ書き込み速度は、光ディスク 8 の所定のファイルが記録されているセクタを読み取っているとき、図 20 (B) に示すように、一定の値 R_{in} になる。一方、データ書き込み速度は、ピックアップ 20 が光ディスク 8 のトラックの間を移動しているとき、または所定のセクタがピックアップ 20 の読み取り可能な位置に来るまで光ディスク 8 の回転を待っているとき（図 20 (B) の時間 T_s の間）、0 になる。

【0225】このため、バッファ 17 へのデータ書き込み速度が 0 になるとき、バッファ 17 に記録されているデータの量は、読み出し速度 R_{out} で読み出しされるだけとなるため、図 20 (A) に示されるように、急激に減少する。バッファ 17 の記憶可能なデータ量は、所定の期間、データ書き込みが無くとも、データの読み出しが途切れないように、 R_{in} 、およびデータの読み出し速度により決定される。

【0226】図 21 は、光ディスク 8 に記録されているファイルの構成を説明する図である。ブロックは、ディスク全体を等しい大きさに分割したもので、ブロック内

が物理的に連続で、かつ、ブロック内ではRinの速度でデータの転送が実行される。ファイルのデータは、1または複数のブロックに記録される。従って、ブロックは、ファイルの一部または全部のデータが記録されているブロック、またはファイルのデータが記録されていないブロックにわかれる。ブロックに記録されているファイルのデータ量がブロックの大きさより小さいとき、そのファイルの直前のブロックは、その全てにデータが記録されている。

【0227】図22は、ファイルの構成とバッファ17に記憶されたデータの量を示す図である。図22(A)は、ブロックに記録されているファイルを説明する図である。ブロック31は、その全てにファイルのデータが記録されている。ブロック31に連続するブロック32は、その一部にファイルのデータが記録されている。ブロック33は、その全てにファイルのデータが記録されている。ブロック33に連続するブロック34は、その一部にファイルのデータが記録されている。

【0228】図22(B)は、図22(A)に示されたブロックを読み出すときのバッファ17への書き込み速度を表す図である。ブロック31を読み出すとき、バッファ17への書き込み速度は、ブロック31が物理的に連続しているため、Rinの一定速度となる。同様に、ブロック32を読み出すとき、ブロック33を読み出すとき、およびブロック34を読み出すとき、バッファ17への書き込み速度は、Rinの一定速度となる。

【0229】ブロック31の読み出しを終了し、つぎにブロック32の読み出しを行うとき、ブロック31とブロック32が物理的に連続しているとは限らないため、連続していなければ、ピックアップ20は、光ディスク8のトラックの間を移動するか、または所定のセクタがピックアップ20の読み取り可能な位置に来るまで光ディスク8の回転を待つ。このため、バッファ17への書き込み速度が0になる期間Ts1が存在する。同様に、ブロック32の読み出しを終了し、つぎにブロック33の読み出しを行うとき、バッファ17への書き込み速度が0になる期間Ts2が存在し、ブロック33の読み出しを終了し、つぎにブロック34の読み出しを行うとき、バッファ17への書き込み速度が0になる期間Ts3が存在する。

【0230】図22(C)は、バッファ17からのデータ読み出し速度を示す図である。データ読み出し速度は、常に一定の値Routである。図22(D)は、バッファ17に記憶されているデータの量を示す図である。図20(A)に示される場合と同様に、バッファ17のデータ量は、書き込み速度Rinと読み出し速度Routの差に対応する速度で増加し、バッファ17へのデータ書き込み速度が0になるとき、バッファ17に記録されているデータの量は、読み出しだけとなるので、急激に減少する。特に、その一部だけにファイルのデータが記録され

ているブロック32およびブロック34を読み出した後のデータ書き込み速度が0になるとき、バッファ17に記録されているデータの量は、大きく減少するため、バッファ17は、アンダフローを防止するには、所定以上の記憶容量が必要となる。

【0231】図23は、光ディスク8に記録されているファイルの他の構成例を説明する図である。この構成では、その一部または全部にファイルのデータが記録されているブロックは、必ず、ブロックの2分の1以上にファイルのデータが記録されるようになされている。

【0232】図24は、ファイルが図23に示すように構成されている場合におけるバッファ17のデータの量の変化を示す図である。図24(A)は、ブロックに記録されているファイルを説明する図である。ブロック51乃至ブロック54は、前述のように、その2分の1以上にファイルが記録されている。

【0233】図24(B)は、図24(A)に示されたブロックを読み出すときのバッファ17への書き込み速度を表す図である。ブロック51を読み出すとき、バッファ17への書き込み速度は、ブロック51が物理的に連続しているため、Rinの一定速度となる。同様に、ブロック52を読み出すとき、ブロック53を読み出すとき、およびブロック54を読み出すとき、バッファ17への書き込み速度は、Rinの一定速度となる。

【0234】ブロック51の読み出しを終了し、つぎにブロック52の読み出しを行うとき、ブロックが物理的に離間していれば、バッファ17への書き込み速度が0になる期間Ts4が存在する。同様に、ブロック52の読み出しを終了し、つぎにブロック53の読み出しを行うとき、バッファ17への書き込み速度が0になる期間Ts5が存在し、ブロック53の読み出しを終了し、つぎにブロック54の読み出しを行うとき、バッファ17への書き込み速度が0になる期間Ts6が存在する。

【0235】図24(C)は、バッファ17からのデータ読み出し速度を示す図である。データ読み出し速度は、常に一定の値Routである。図24(D)は、バッファ17に記憶されているデータの量の変化を示す図である。バッファ17へのデータ書き込み速度が0になるとき、バッファ17に記録されているデータの量は、急激に減少する。図22(D)の場合と比較し、ブロック51、ブロック52、ブロック53、およびブロック54は、一定量(1/2)以上のデータを記録しているため、バッファ17に記録されているデータの量が、0に近く可能性は、図22(D)に示した場合より、少ない。

【0236】図25は、ファイル管理部6のファイルのブロックへの記録の処理を説明する図である。図25(A)に示すように、すでにブロック71乃至73にファイルのデータが記録されており、新たに、ブロック74にブロック74の2分の1より小さいデータ量のファイル75が記録される場合の処理を説明する。図25

(B) に示すように、ブロック 73 に記憶されたファイルは、ブロック 73 の 2 分の 1 を占める前半部分 81 を残して分割され、後半部分 82 がブロック 74 の先頭に移動される。ファイル 75 は、ブロック 74 の後半部分 82 に続いて記録される。

【0237】 以上のように、ファイルの一部または全部が記録されているブロックは、ブロックの 2 分の 1 以上にファイルが記録される。

【0238】 以上の処理をまとめると、図 26 のフローチャートに示すようになる。すなわち、ステップ S31 において、ファイル管理部 6 は、記録するデータ量がブロックの 1/2 未満であるか否かを判定し、記録するデータ量がブロックの 1/2 未満であると判定された場合、ステップ S32 に進み、直前のブロックの後方の 1/2 のデータを分割し、次のブロックに記録させる。ステップ S33 において、ファイル管理部 6 は、そのブロックにブロックの 1/2 未満の量のデータを記録する。

【0239】 ステップ S34 において、ファイル管理部 6 は、全てのデータを記録したか否かを判定し、全てのデータを記録していないと判定された場合、ステップ S31 に戻り、処理を繰り返す。

【0240】 ステップ S31 において、記録するデータ量がブロックの 1/2 未満でないと判定された場合、ステップ S35 に進み、ファイル管理部 6 は、記録するデータ量が 1 ブロック分以下であるか否かを判定し、記録するデータ量が 1 ブロック分以下でないと判定された場合、ステップ S36 に進む。ステップ S36 において、ファイル管理部 6 は、1 ブロック分のデータを記録し、ステップ S34 に進む。

【0241】 ステップ S35 において、記録するデータ量が 1 ブロック分以下であると判定された場合、ステップ S37 に進み、ファイル管理部 6 は、そのデータを 1 ブロックに記録し、ステップ S34 に進む。

【0242】 ステップ S34 において、全てのデータを記録したと判定された場合、処理は終了する。

【0243】 図 27 は、ブロックへの記録されたファイルの分割の処理を説明する図である。図 27 (A) に示すように、ブロック 91 乃至 93 に 1 つのファイルが記録されており、このファイルをブロック 91 の始点からブロック 92 の分割点 (ブロック 92 の 1/2 より前方に位置する) までのファイルと、ブロック 92 の分割点からブロック 93 の終点までのファイルに分割する場合の処理を説明する。図 27 (B) に示すように、ブロック 92 の始点から分割点までの部分 95 の前の部分が記憶されたブロック 91 のデータは、2 分割され、その後半部分 94 は、ブロック 92 に移動される。ブロック 92 に移動された後半部分 94 に続いて、ブロック 92 の前半部分 95 が格納される。一方、ブロック 92 の分割点から終点までの部分 96 は、新たなブロック 101 に格納される。

【0244】 図 28 は、ファイルの分割の他の処理例を説明する図である。図 28 (A) に示すように、ブロック 111 乃至 114 に記録されている 1 つのファイルを、ブロック 112 の 1/2 の位置より前に位置する分割点で分割する場合の処理を説明する。

【0245】 図 28 (B) に示すように、ブロック 111 に、ブロック 112 の始点から分割点までの部分 115 を記録できる大きさの領域があれば、ブロック 111 のすでに記録されているファイルに続いて、部分 115 が記録される。ブロック 112 の始点からデータの最後までの部分 116 は、ブロック 112 の始点からの位置に移動される。全ての範囲に記録されたブロック 113 のデータは 2 分割され、その前半部分 117 は、ブロック 112 に移動され、ブロック 112 の部分 116 に続いて記録される。ブロック 113 の後半部分 118 は、ブロック 113 の始点からの位置に移動される。

【0246】 図 29 は、ブロックへの記録されたファイルの分割のさらに異なる処理例を説明する図である。図 29 (A) に示すように、ブロック 121 乃至 123 に記録されている 1 つのファイルを、ブロック 122 の中間点を分割点として分割する場合の処理を説明する。図 29 (B) に示すように、ブロック 122 の分割点からデータの最後までの部分 124 は、新たなブロック 131 の先頭に格納される。全ての範囲に記録されたブロック 123 のファイルは 2 分割され、その前半部分 125 は、ブロック 131 に、部分 124 に続いて格納され、後半部分 126 は、ブロック 123 の先頭に移動される。

【0247】 以上のように、ファイルが分割されても、ブロックは、その 2 分の 1 以上にファイルが記録される。

【0248】 図 27 に示された、ブロックの始点から分割点までのデータの大きさがブロックの大きさの 1/2 未満であり、かつ、分割点から後ろのデータの大きさがブロックの大きさの 1/2 以上である場合のファイルの分割の処理は、図 30 のフローチャートに示すようになる。すなわち、ステップ S41 において、ファイル管理部 6 は、分割点のあるブロックの、分割点から後ろのデータを、新たなブロックに移動する。ステップ S42 において、ファイル管理部 6 は、分割点のあるブロックの直前のブロックの所定のデータを、分割点のあるブロックの始点からの位置に移動し、分割点のあるブロックの始点から分割点までデータをそのデータの後ろに移動する。

【0249】 図 28 に示された、分割点のあるブロックの直前の空きの大きさが、分割点のあるブロックの始点から分割点までのデータの大きさ以上であり、かつ、分割点から後ろのデータの大きさがブロックの大きさの 1/2 未満である場合のファイルの分割の処理は、図 31 のフローチャートに示すようになる。ステップ S51 において、ファイル管理部 6 は、分割点のあるブロックの、

ブロックの始点から分割点までデータを、分割点のあるブロックの直前のブロックの空きに移動する。ステップS52において、ファイル管理部6は、分割点のあるブロックの直後のブロックの所定のデータを、分割点のあるブロックのデータの後ろに移動する。

【0250】図29に示された、ブロックの始点から分割点までのデータの大きさがブロックの大きさの1/2以上であり、かつ、分割点から後ろのデータの大きさがブロックの大きさの1/2未満である場合のファイルの分割の処理は、図32のフローチャートに示すようになる。ステップS61において、ファイル管理部6は、分割点から後ろのデータを、新たなブロックに移動する。ステップS62において、ファイル管理部6は、分割点のあるブロックの直後のブロックの所定のデータを、新たなブロックのデータの後ろの位置に移動する。

【0251】以上においては、ブロックの始点から分割点までのデータの大きさがブロックの大きさの1/2以上であるか否かを基準としたが、 $(n-1)/n$ ($n=2, 3, 4, 5, \dots$) を基準としてもよい。

【0252】図33は、連続する3つのブロックの空き領域が、合わせて1ブロック以上ある場合の、ブロックの空き領域の圧縮の処理を説明する図である。図33

(A)に示すように、ブロック141乃至143の空き領域は、合わせて1ブロック以上ある。ブロック142に記憶された内容を、ブロック141の空き領域の大きさと同じ大きさの部分144と残りの部分145に分割する。

【0253】図33(B)に示すように、ブロック142の部分144はブロック141の空き領域に移動される。ブロック142の部分145は、ブロック142の先頭に移動され、ブロック143のデータ146は、ブロック142に移動され、部分145に続けて格納される。ブロック143は、空きになる。

【0254】このように、ブロック141乃至142の空き領域を少なくし、ブロック143を空きにすることができる。

【0255】以上の処理をまとめると、図34のフローチャートになる。すなわち、ステップS71において、ファイル管理部6は、3つのブロックの空きの合計が、1ブロック以上であるか否かを判定し、3つのブロックの空きの合計が、1ブロック以上であると判定された場合、ステップS72に進み、中間のブロックから、先頭のブロックの空きに、その空きの相当するデータを移動する。ステップS73において、ファイル管理部6は、最後のブロックから、中間のブロックの空きに、その空きの相当するデータを移動し、処理を終了する。

【0256】ステップS71において、3つのブロックの空きの合計が、1ブロック以上でないと判定された場合、処理は終了する。

【0257】以上のように、ファイルの一部または全部

が記録されているブロックは、ブロックの2分の1以上にファイルが記録され、書き込み速度が0になる期間が分散されるため、バッファ17の容量が少なくても出力が途切れない。

【0258】なお、上記したような処理を行うコンピュータプログラムをユーザに提供する提供媒体としては、磁気ディスク、CD-ROM、固体メモリなどの記録媒体の他、ネットワーク、衛星などの通信媒体を利用することができる。

【0259】

【発明の効果】請求項1に記載の記録再生装置、請求項3に記載のファイル管理方法、および請求項4に記載の提供媒体によれば、AVデータのファイルを記録し、管理情報を、論理ボリウムの少なくとも2個所に記録するようにしたので、個人が家庭内で圧縮ビデオ、圧縮音声信号を簡単に記録再生することができる。

【0260】請求項5に記載の記録再生装置、請求項9に記載のファイル管理方法、および請求項10に記載の提供媒体によれば、ディスク状記録媒体に対して記録可能な情報の単位を記録し、記録する情報の単位の長さを設定し、ディスク状記録媒体に記録されるファイルに対応して、設定された長さの単位のうち、そのファイルを記録するとき採用された単位の長さを識別するようにしたので、個人が家庭内で圧縮ビデオ、圧縮音声信号を簡単に記録再生することができる。

【図面の簡単な説明】

【図1】ディスク状記録媒体全体のフォーマットを説明する図である。

【図2】アンカーディスクリプタについて説明する図である。

【図3】ロジカルボリウムを説明する図である。

【図4】アロケーションエクステントの長さの設定の処理を説明するフローチャートである。

【図5】アロケーションエクステントの長さを設定する画面の例を示す図である。

【図6】ファイルの記録の処理を説明するフローチャートである。

【図7】アロケーションエクステントの長さを選択する画面の例である。

【図8】ボリウムストラクチャディスクリプタを説明する図である。

【図9】メディアインフォメーションディスクリプタを説明する図である。

【図10】ドライブインフォメーションディスクリプタを説明する図である。

【図11】エクステンデッドデータディスクリプタを説明する図である。

【図12】ファイルシステムを説明する図である。

【図13】チャイルドリンク、ネクストリンク、およびペアレントリンクを説明する図である。

【図14】ファイルテーブルを説明する図である。

【図15】ファイルテーブルストラクチャタイプ0のファイルテーブルを説明する図である。

【図16】アロケーションエクステンツテーブルを説明する図である。

【図17】アロケーションストラテジテーブルを説明する図である。

【図18】エクステンデッドアトリビュートテーブルを説明する図である。

【図19】本発明の記録再生装置1の一実施の形態の構成を示すブロック図である。

【図20】再生のときの、バッファ17に記録されているデータの量とバッファ17に書き込まれるデータの速度の関係を示す図である。

【図21】光ディスク8に記録されているファイルの構成を説明する図である。

【図22】ファイルの構成とバッファ17に記憶されたデータの量を示す図である。

【図23】光ディスク8に記録されているファイルの他の構成を説明する図である。

【図24】図23の場合のファイルの構成とバッファ17に記憶されているデータの量を示す図である。

【図25】ファイルのブロックへの記録の処理を説明する図である。

【図26】ブロックへのデータの記録の処理を説明する

フローチャートである。

【図27】ブロックへの記録されたファイルの分割の処理を説明する図である。

【図28】ブロックへの記録されたファイルの分割の他の処理を説明する図である。

【図29】ブロックへの記録されたファイルの分割のさらに異なる処理を説明する図である。

【図30】ファイルの分割の処理を説明するフローチャートである。

【図31】ファイルの分割の処理を説明するフローチャートである。

【図32】ファイルの分割の処理を説明するフローチャートである。

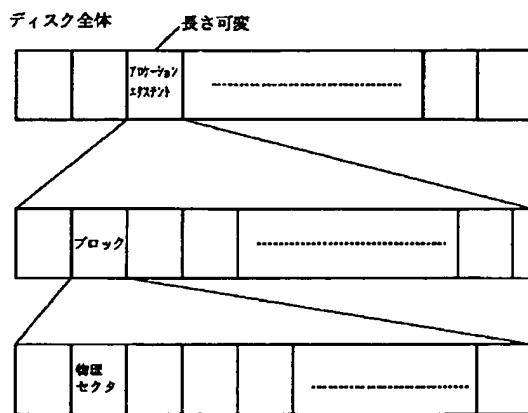
【図33】ブロックの空き領域の圧縮の処理を説明する図である。

【図34】ブロックの空きの圧縮の処理を説明するフローチャートである。

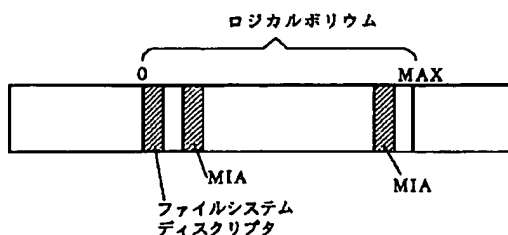
【符号の説明】

1 記録再生装置, 2 ユーザ入出力部, 3 AV入出力部, 4 PCデータ入出力部, 5 システムコントロール部, 6 ファイル管理部, 7 ドライブ部, 8 光ディスク, 31乃至34, 51乃至54, 71乃至74, 91乃至93, 101, 111乃至114, 121乃至123, 131, 141乃至143
ブロック

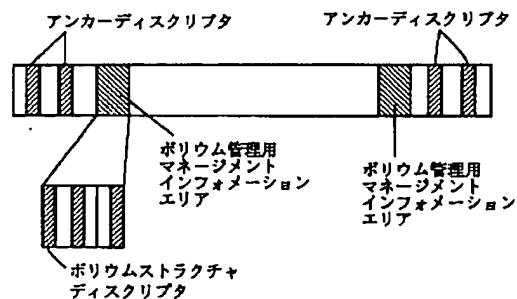
【図1】



【図3】



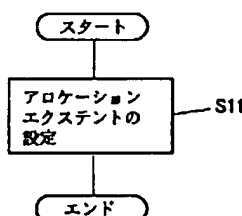
【図2】



【図9】

```
[Media Information Descriptor] {
  <Media Information Descriptor Header>
  <Zone Information> 0+
}
```

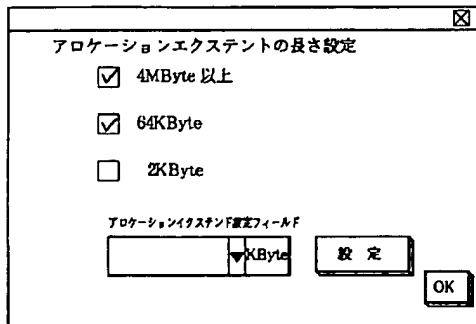
【図4】



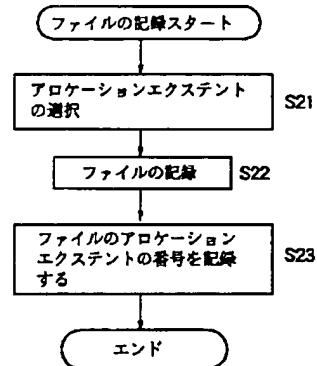
【図10】

```
[Drive Information Descriptor] {
  <Drive Information Descriptor Header>
}
```

【図 5】



【図 6】



【図 11】

```

[Extended Data Descriptor] {
  <Extended Data Descriptor Header>
  [Extended Data Set] {
    @APS <Extended Data> 0+
  }
}

```

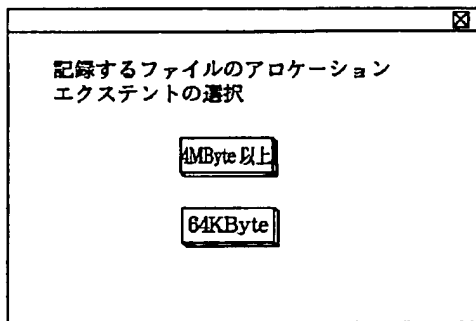
【図 14】

```

[File table] {
  <File Table Header>
  <File Table Data>
}

```

【図 7】



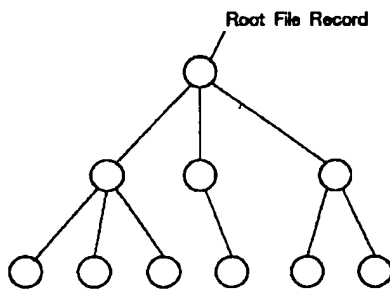
【図 8】

```

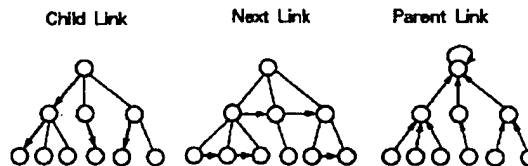
<Spare Area Information> 0+
[Logical Volume Information] {
  <Logical Volume Information Header>
  <Partition Map> +1
} 1+
[Defect List Information] {
  <Defect List Information Header>
  [Defect Lists] {
    @APS <Primary Defect List> 0+1
    @APS <Secondary Defect List> 0+1
  } 1+
} 0+1

```

【図 12】



【図 13】



【図 15】

```

[File Table] {
  <File Table Header>
  <File Record> 1+
}

```

【図 16】

```

[Allocation Extents Table] {
  <Allocation Extents Table Header>
  <Allocation Extent Record> 0+
}

```

【図 17】

```

[Allocation Strategy Table] {
  <Allocation Strategy Table Header>
  <Allocation Strategy Record> 1+
}

```

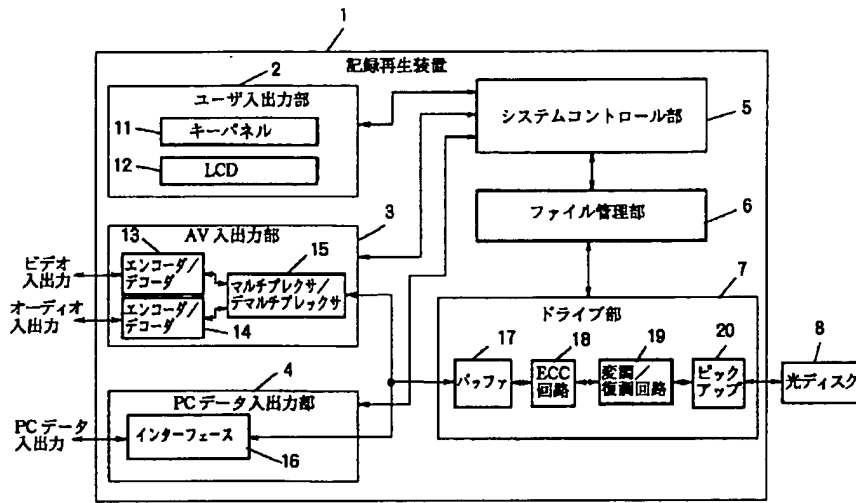
【図 18】

```

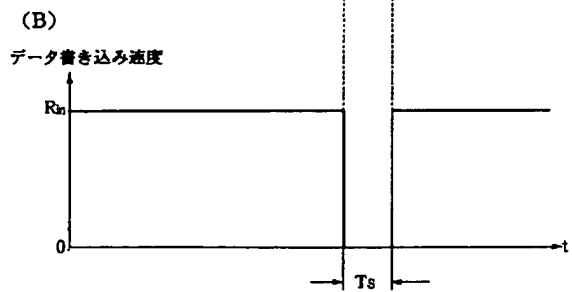
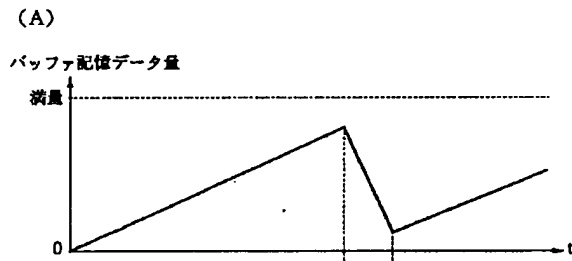
[Extended Attribute Table] {
  <Extended Attribute Table Header>
  <Extended Attribute Record> 1+
}

```

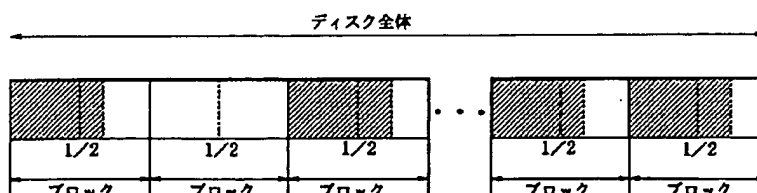
【図19】



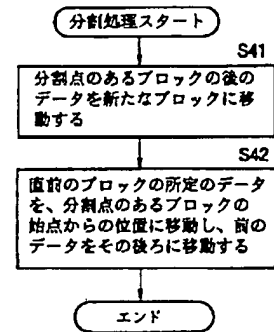
【図20】



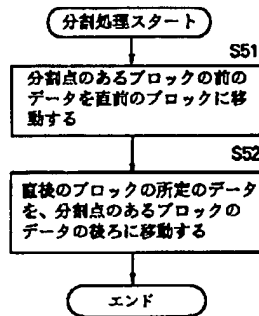
【図23】



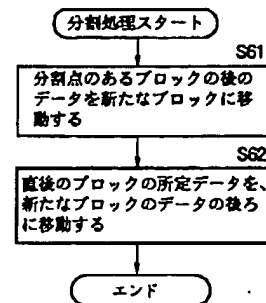
【図30】



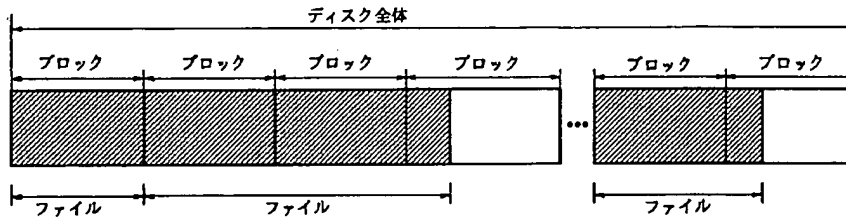
【図31】



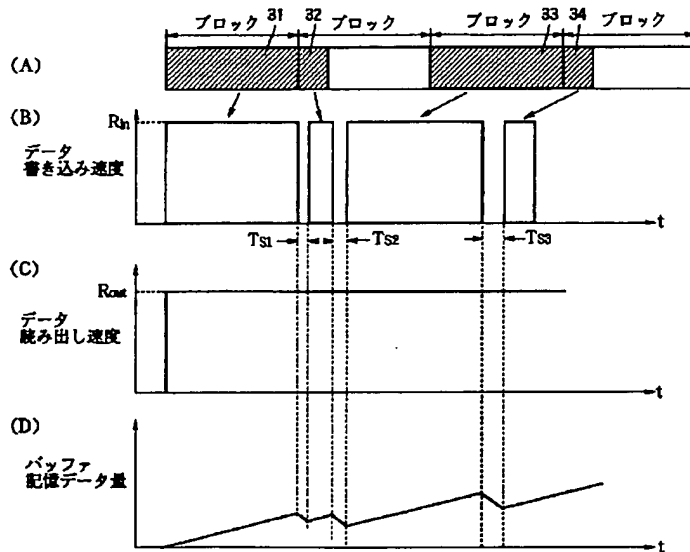
【図32】



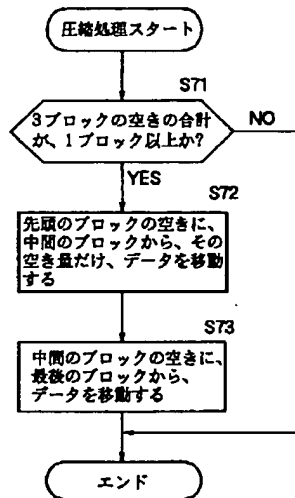
【図 21】



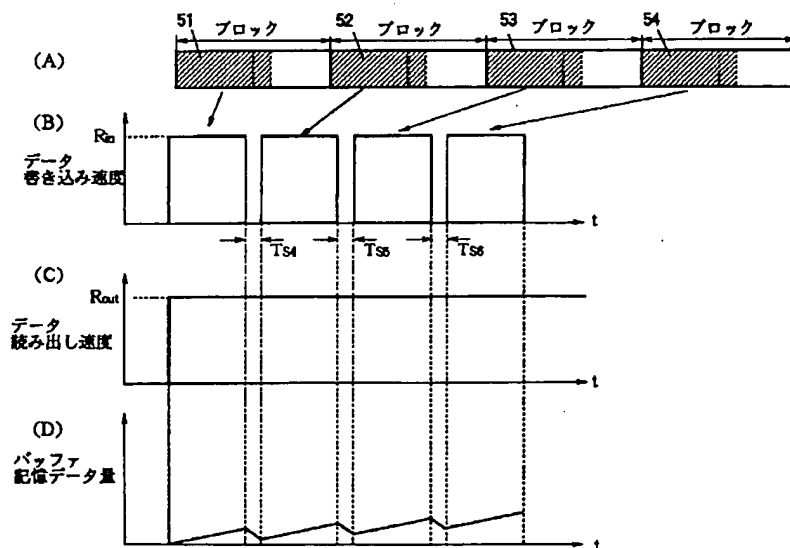
【図 22】



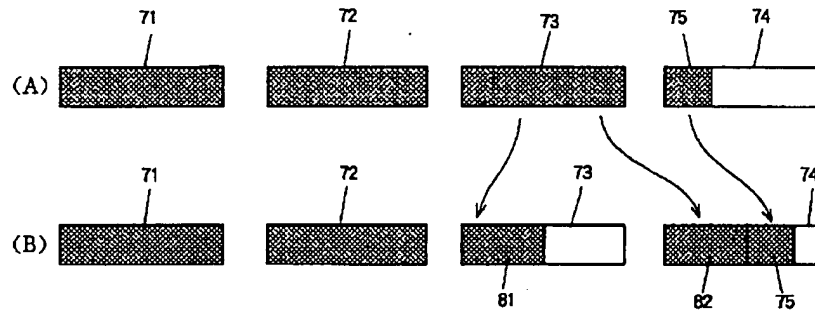
【図 34】



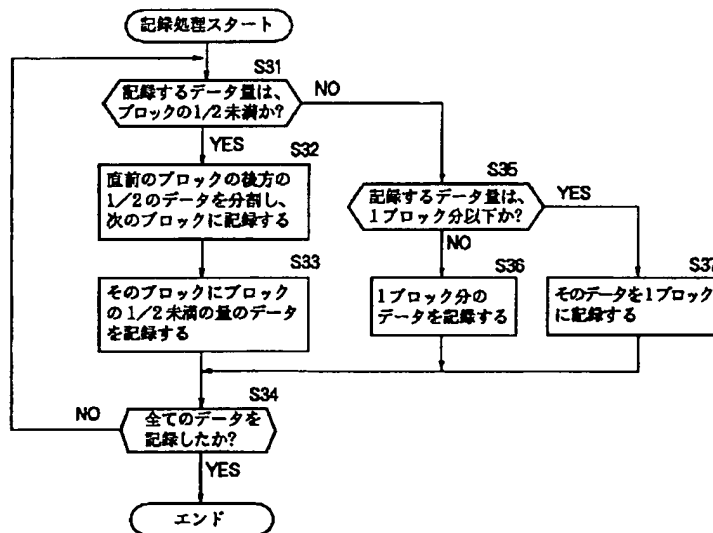
【図 24】



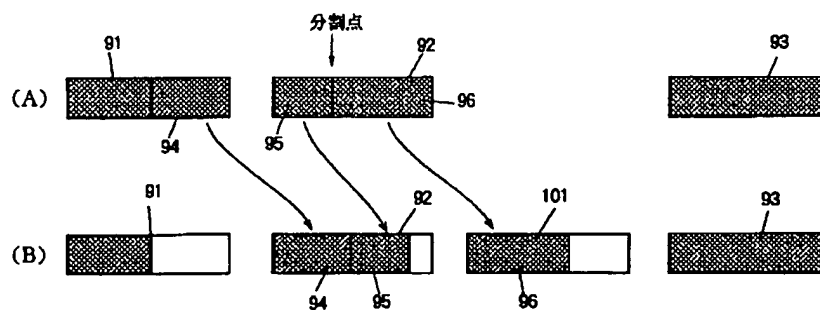
【図 2 5】



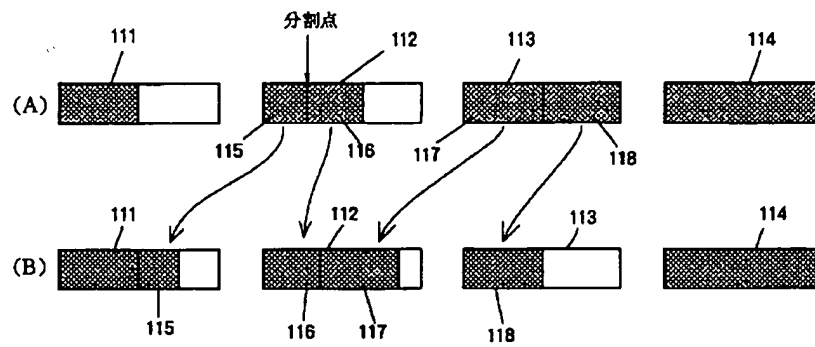
【図 2 6】



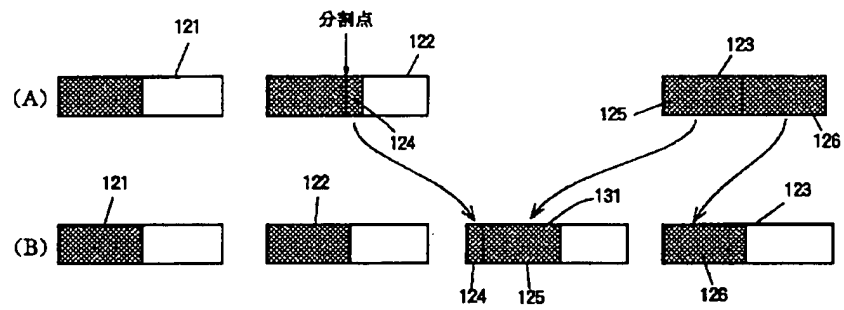
【図 2 7】



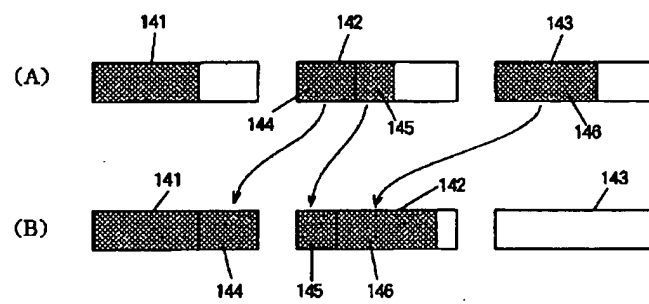
【図 2 8】



【図 2 9】



【図 3 3】



JP-11-312378

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The record regenerative apparatus characterized by having the 1st record means which records the file of AV data, and the 2nd record means which records the management information of the aforementioned file on at least two places of logic bohrium in the record regenerative apparatus which records or reproduces AV data to a disk-like record medium.

[Claim 2] The record regenerative apparatus according to claim 1 characterized by including the information about a defective sector and an intact sector at least at the management information recorded on the aforementioned record means.

[Claim 3] The file management method characterized by including the 1st record step which records the file of AV data, and the 2nd record step which records the management information of the aforementioned file on at least two places of logic bohrium in the file management method of the record regenerative apparatus which records or reproduces AV data to a disk-like record medium.

[Claim 4] The offer medium by which the computer which performs processing containing the 1st record step which records the file of AV data on the record regenerative apparatus which records or reproduces AV data to a disk-like record medium, and the 2nd record step which records the management information of the aforementioned file on at least two places of logic bohrium is characterized by offering the program in which a readout is possible.

[Claim 5] The record regenerative apparatus characterized by providing the following. A unit record means to record the unit of recordable information to a disk-like record medium. A setting means to set up the length of the unit of the information recorded by the aforementioned unit record means. An identification information record means to record the identification information which discriminates the length of the aforementioned unit adopted when recording the file among the units of length set up by the aforementioned setting means corresponding to the file recorded on the aforementioned disk-like record medium.

[Claim 6] The aforementioned setting means is a record regenerative apparatus according to claim 5 characterized by setting up the length of the unit of AV data for a long time than the length of the unit of computer data.

[Claim 7] The record regenerative apparatus according to claim 5 characterized by having further the control means controlled so that the aforementioned disk-like record medium is divided into two or more blocks and data are recorded on 1/2 or more fields of the divided aforementioned block.

[Claim 8] The record regenerative apparatus according to claim 5 characterized by having further the control means controlled so that the truck which records information on the aforementioned disk-like record medium is divided into two or more blocks and data are recorded on the field (n-1) / more than n of the divided aforementioned block.

[Claim 9] The file management method of the record regenerative apparatus which records or reproduces information to a disk-like record medium characterized by providing the following. The unit record step which records the unit of recordable information to a disk-like record medium. The setting step which sets up the length of the unit of the information recorded at the aforementioned unit record

step. The identification information record step which records the identification information which discriminates the length of the aforementioned unit adopted when recording the file among the units of length set up at the aforementioned setting step corresponding to the file recorded on the aforementioned disk-like record medium.

[Claim 10] The offer medium by which the computer which is characterized by providing the following, and which performs processing is characterized by offering the program in which a readout is possible. The unit record step which records the unit of recordable information on the record regenerative apparatus which records or reproduces information to a disk-like record medium to the aforementioned disk-like record medium. The setting step which sets up the length of the unit of the information recorded at the aforementioned unit record step. The identification information record step which records the identification information which discriminates the length of the aforementioned unit adopted when recording the file among the units of length set up at the aforementioned setting step corresponding to the file recorded on the aforementioned disk-like record medium.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[The technical field to which invention belongs] Especially this invention relates to the file system used for the record regenerative apparatus (VDR: Video Disc Recorder) which uses a disk-like record medium about a record regenerative apparatus, the file management method, and an offer medium.

[0002]

[Description of the Prior Art] As a file system for recording data on a disk-like record medium ISO/IEC 13346:1995 and "Information technology-Volume and file structure of write-once and rewritable media using non-sequential recording for information interchange." is known. This file system is a general-purpose file system for recording various data, and there is at the thing for recording the digital AV (voice, video) signal with which the individual was compressed into the disk by domestic. [no] Therefore, it is not necessarily enough to record the compressed digital AV (voice, video) signal. Therefore, the optimal file system for recording AV signal and bohrium are called for.

[0003]

[Problem(s) to be Solved by the Invention] Therefore, the file system for carrying out record reproduction of the AV signal is simply required for a disk domestic in an individual.

[0004]

[Means for Solving the Problem] A record regenerative apparatus according to claim 1 is characterized by having the 1st record means which records the file of AV data, and the 2nd record means which records management information on at least two places of logic bohrium.

[0005] The file management method according to claim 3 is characterized by including the 1st record step which records the file of AV data, and the 2nd record step which records management information on at least two places of logic bohrium.

[0006] It is characterized by an offer medium according to claim 4 offering the program in which the readout of the computer which performs processing containing the 1st record step which records the file of AV data, and the 2nd record step which records management information on at least two places of logic bohrium is possible.

[0007] A unit record means by which a record regenerative apparatus according to claim 5 records the unit of recordable information to a disk-like record medium, A setting means to set up the length of the unit of the information recorded by the unit record means, It is characterized by having an identification information record means to record the identification information which discriminates the length of the unit adopted when recording the file among the units of length set up by the setting means corresponding to the file recorded on a disk-like record medium.

[0008] The unit record step on which the file management method according to claim 9 records the unit of recordable information to a disk-like record medium, The setting step which sets up the length of the unit of the information recorded at a unit record step, It is characterized by including the identification information record step which records the identification information which discriminates the length of the unit adopted when recording the file among the units of length set up at the setting step

corresponding to the file recorded on a disk-like record medium.

[0009] The unit record step on which an offer medium according to claim 10 records the unit of recordable information to a disk-like record medium, The setting step which sets up the length of the unit of the information recorded at a unit record step, The inside of the unit of length set up at the setting step corresponding to the file recorded on a disk-like record medium, The computer which performs processing containing the identification information record step which records the identification information which discriminates the length of the unit adopted when recording the file is characterized by offering the program in which a readout is possible.

[0010] In a record regenerative apparatus according to claim 1, the file management method according to claim 3, and an offer medium according to claim 4, the file of AV data is recorded and management information is recorded on at least two places of logic bohrium.

[0011] In a record regenerative apparatus according to claim 5, the file management method according to claim 9, and an offer medium according to claim 10 The length of the unit adopted when recording the file among the units of length which set up the length of the unit of the information which records and records the unit of recordable information to a disk-like record medium, and were set up corresponding to the file recorded on a disk-like record medium is discriminated.

[0012]

[Embodiments of the Invention] Although the gestalt of operation of this invention is explained below, it is as follows, when the gestalt (however, an example) of operation [in the parenthesis after each means] is added and the feature of this invention is described, in order to clarify correspondence relation between each means of invention given in a claim, and the gestalt of the following operations. However, of course, this publication does not mean limiting to what indicated each means.

[0013] That is, a record regenerative apparatus according to claim 1 is characterized by having the 1st record means (for example, drive section 7 of drawing 19) which records the file of AV data, and the 2nd record means (for example, drive section 7 of drawing 19) which records management information on at least two places of logic bohrium.

[0014] A unit record means by which a record regenerative apparatus according to claim 5 records the unit of recordable information to a disk-like record medium (for example, MIA of drawing 3), A setting means to set up the length of the unit of the information recorded by the unit record means (for example, step S11 of processing of drawing 4), The inside of the unit of length set up by the setting means corresponding to the file recorded on a disk-like record medium, It is characterized by having an identification information record means (for example, step S23 of processing of drawing 6) to record the identification information which discriminates the length of the unit adopted when recording the file.

[0015] A record regenerative apparatus according to claim 7 is characterized by having further the control means (for example, file management section 6 of drawing 19) controlled so that the aforementioned disk-like record medium is divided into two or more blocks and data are recorded on 1/2 or more fields of the divided aforementioned block.

[0016] First, the format method of a disk-like record medium is explained. Drawing 1 is drawing explaining the format of the whole disk-like record medium. A disk is divided into the allocation extent of the plurality [length] of variable length. An allocation extent consists of two or more blocks of fixation of length. A block consists of a predetermined number of physical sectors.

[0017] Drawing 2 is drawing explaining a support descriptor. Four support descriptors are arranged in a disk. The position of bohrium administrative management information area is recorded on the support descriptor. The physical bohrium information, the party SHON information, the logical bohrium information, and the party SHON map are contained in the bohrium structure descriptor of bohrium administrative management information area.

[0018] The logical bohrium as user area is described by the bohrium structure descriptor. Drawing 3 is drawing explaining logical bohrium. The file system descriptor is arranged at logical bohrium. Near a head and near an end logical bohrium, MIA (Management Information Area) is arranged, respectively. A file table, the allocation extent table, the allocation strike RATEJII table, the defect information table,

and the EKUSUTEN dead attribute table are contained in MIA. The length of an allocation extent is described by the allocation strike RATEJII record which constitutes an allocation strike RATEJII table. [0019] A user sets up beforehand the length of the allocation extent of the data recorded on the disk, before recording the data of a file on a disk. Thereby, for example, AV data are recorded by the format of the allocation extent of longer length, and are that PC data can be recorded by the format of the allocation extent of short length. Since AV data are continuous data in many cases, the direction which lengthened the length of an allocation extent can carry out record reproduction of the data more efficiently.

[0020] Drawing 4 is a flow chart explaining processing of a setup of the length of an allocation extent. In Step S11, the drive section 7 mentioned later writes in the allocation strike RATEJII record corresponding to the length by which the allocation extent was set as the allocation strike RATEJII table contained in MIA corresponding to the setting input from a user. Two or more allocation strike RATEJII records can be written in an allocation strike RATEJII table. Drawing 5 is drawing showing the example of the screen where a user sets up the length of an allocation extent. As length of an allocation extent, a setup of the length of plurality [set / arbitrary length, such as 4 or more MBytes 64KByte, and 2kByte, / and] is possible. It was able to be specified out of the format of the allocation extent of the length set up beforehand, can be accepted on the disk, and can record on it.

[0021] Thus, after setting up the length of an allocation extent and recording on a disk, the processing in the case of recording data on the disk comes to be shown in the flow chart of drawing 6 . In Step S21, a user chooses the length of the allocation extent of the data to be recorded from now on. Drawing 7 expresses the example of the screen which chooses the length of an allocation extent. As this length, only the value beforehand set as the disk is displayed. By operating the button of a screen, the length of the allocation extent corresponding to a button is chosen. When recording AV data, compared with the time of recording PC data, record of more efficient data is attained by specifying a longer allocation extent. By specification of the length of an allocation extent, the allocation strike RATEJII record arranged in an allocation strike RATEJII table is specified. If specification is completed, in Step S22, the drive section 7 will record the inputted data on a disk. If record of data is completed, in Step S23, the drive section 7 will record the number corresponding to the length of the allocation extent of the file on a disk. The file management section 6 mentioned later can use the content of a corresponding allocation strike RATEJII record by knowing the number corresponding to the length of an allocation extent.

[0022] When the system-control section 5 of drawing 19 mentioned later can judge whether it is going to record PC data for whether it is going to record AV data, it is also possible to perform the above-mentioned step S21 without the input from a user.

[0023] As mentioned above, a file is recorded on a disk.

[0024] The composition of bohrium is explained. A disk extent (DescExtent) is used for expressing the field by which alignment was carried out to MIB (Management Information Block) later mentioned in the descriptor (descriptor) recorded in the below-mentioned MIA. A disk extent is recorded in the format shown in Table 1.

[0025]

[Table 1]

Desc Extent

RBP	Length	Name	Contents
0	2	Offset (Number of MIB) from top of a descriptor	Uint 16
2	2	Length (Number of MIB)	Uint 16

[0026] An offset FUROMUTOPPUOBU descriptor (Offset from top of a descriptor:RBP 0) specifies the offset (the number of MIB) from the head MIB of a descriptor (descriptor) to a field. Length (Length:RBP 2) specifies the size (the number of MIB) of a field.

[0027] A PDL entry (Primary Defect List Entry) is used for recording the physical sector size (physical sector size) of the physical sector (physical sector) which performs slipping (slipping) in defect

management (defect management). A PDL entry is recorded in the format shown in Table 2.

[0028]

[Table 2]

PDL Entry

RBP	Length	Name	Contents
0	4	Physical Sector Number of Defect Sector	Uint 32

[0029] A physical SEKUTANAMBAOBU defect sector (Physical Sector Number of Defect Sector:RBP 0) specifies the physical sector number of the physical sector which performs slipping.

[0030] An SDL entry (Secondary Defect List Entry) is used for recording the physical sector number of a linear replacement MENTO **** physical sector, and the physical sector number of the physical sector used as the alternative in defect management. An SDL entry is recorded in the format shown in Table 3.

[0031]

[Table 3]

SDL Entry

RBP	Length	Name	Contents
0	4	Physical Sector Number of Defect Sector	Uint 32
4	4	Physical Sector Number of Spare Sector	Uint 32

[0032] The physical SEKUTANAMBAOBU defect sector of Table 3 specifies the physical sector number of the physical sector which performs linear replacement MENTO. A physical SEKUTANAMBAOBU spare sector (Physical Sector Number of Spare Sector:RBP 4) specifies the physical sector number of the alternative physical sector used by linear replacement MENTO.

[0033] An anchor point (Anchor points) is the start point of volume structural analysis. A support descriptor (Anchor Descriptor) is recorded on an anchor point. The physical sector number of the physical sector which is an anchor point is not specified.

[0034] However, it is prescribed as follows by VDR. That is, in the case of a ROM (Read Only Memory) disk and a RAM (Random Access Memory) disk, let Ch, 20 h, LPSN (Last Physical Sector Number)-20h, and LPSN-Ch (the numeric value which h attached at the end expresses a hexadecimal) be anchor points. In the case of a partial ROM disk, let Ch in the field of ROM and each RAM, 20 h, LPSN-20h, and LPSN-Ch be anchor points. In this case, when the suitable information for the anchor point of a RAM field is recorded, it is used, and when suitable information is not recorded, the information on a ROM field is used.

[0035] A support descriptor is recorded on the physical sector which is an anchor point from the byte position 0. The size of a support descriptor is below a physical sector size. Moreover, the field from the next byte of the last byte of a descriptor to the byte of the last of the physical sector is reserved for the future extension, and assigns #00 to all bytes. The definition of a main (Main) MIA field, the definition of a reserve (Reserve) MIA field, the positional information of each MIA map (Map), etc. are recorded on a support descriptor.

[0036] Various kinds of information about bohrium is bohrium administrative management information area. (MIA) It is recorded. Because of reliability reservation, MIA with the information on an equal content is recorded on two on physical bohrium, and it is Main MIA, respectively. Reserve MIA is called. The physical sector in MIA is called management information block (Management Information Block:MIB), and the offset from the head MIB of MIA of the physical sector number is called a management information block number (Management Information Block Number:MIB Number). An MIB number is used for specification of MIB. MIB which cannot use MIA according to a defect etc.,

intact MIB, And the MIA map of Main MIA (MIA Map for Main MIA), The MIA map of Reserve MIA (MIA Map for Reserve MIA), A bohrium structure descriptor (Volume Structure Descriptor), A media information descriptor (Media Information Descriptor), A drive information descriptor (Drive Information Descriptor), And it consists of MIB used to record the data of an extent data descriptor (Extended Data Descriptor).

[0037] It is recorded on a MIA map for which purpose MIB in MIA is used. Main MIA, the starting position of Reserve MIA, and the position of a size and the MIA map in MIA are prescribed by the support descriptor. The above-mentioned data may be recorded over two or more MIB, when recorded in one MIB. When data are recorded on two or more MIB, it is recorded on the Map entry (Map Entries) field in a MIA map which MIB is connected in which turn. As data were MIB, when it finishes, the byte of the last of the MIB sets up #00 from the next byte of the end of data.

[0038] Below, a partition (Partition) is explained. The data storage area defined by the partition information (Partition Information) in a bohrium structure descriptor (Volume Structure Descriptor) is called a partition (partition). One physical bohrium can be divided into two or more partitions. The number for specifying a partition within physical bohrium is called a partition number. A partition number is an integer which begins from 0 and increases every [1] in monotone. All the physical sectors in the same partition are the same physical sector sizes.

[0039] A partition is defined as a table of a partition information (Partition Information) in a bohrium structure descriptor. A partition information defines a partition by the physical sector number of the physical sector of the head of a partition, and the number of the physical sectors belonging to the partition. In physical bohrium, one or more partitions are surely defined. A partition number is determined in the sequence that the partition information was recorded on the bohrium structure descriptor. The partition number of the partition defined by the 1st partition information is 0, the 2nd is 1, they increase every [1] henceforth, and the n-th is n-1.

[0040] Logic bohrium (Logical volume) is explained below. Logic bohrium means the data storage area defined as a meeting of a partition in the logic bohrium information (Logical Volume Information) of a bohrium structure descriptor. The field of logic bohrium connects a partition field in order of description of the partition map (Partition Map) of a logic bohrium information, and is constituted. A partition map specifies the partition which belongs to logic bohrium in the group of the partition number in the bohrium identifier (Volume Identifier) which sets physical bohrium to a meaning, and its physical bohrium. Logic bohrium may consist of partitions belonging to different physical bohrium, and one partition may belong to two or more logic bohrium.

[0041] Logic bohrium is treated as one field regardless of a division break, a physical sector, etc. of a partition, and the content is written per logical sector. A logical sector number is an integer which begins from 0 and increases every [1] in monotone. When the size of logic bohrium is not the multiple of a logic sector size, the odd field produced in the last physical sector is reserved for the future extension, and is not used. An informational definition, a definition of logic bohrium, etc. about the partition with which a bohrium structure descriptor is contained in the physical bohrium are described. When defining the logic bohrium over two or more physical bohrium, it is surely a partition number. A logic bohrium information is described by the bohrium structure descriptor of the physical bohrium with which the partition of 0 is defined.

[0042] In addition, it is a partition number because of reliability reservation. You may describe a logic bohrium information to the bohrium structure descriptor of the physical bohrium with which partitions other than zero belong. A bohrium structure descriptor is recorded on MIA.

[0043] Defect management (Defect management) is explained below. Defect management according to slipping and linear replacement MENTO the whole partition is possible. Specification of whether to perform defect management to each partition is performed at the partition information of a bohrium structure descriptor. The alternative data area used for slipping and linear replacement MENTO is called spare area (spare area). In the partition which performs defect management, and the partition belonging to the same logic bohrium, the above one spare area is surely secured. Moreover, when performing linear replacement MENTO, the last of the partition field serves as spare area.

[0044] When performing slipping, the head portion of the spare area secured in the last of the partition field is used as spare area. Moreover, when performing linear replacement MENTO, as long as an alternative data area is a partition which belongs to the same logic bohrium and belongs to the same physical bohrium, it may use spare area other than the spare area in a partition with a defect sector (defect sector).

[0045] The information about slipping and linear replacement MENTO is recorded on the defect list information (Defect List Information) of a bohrium structure descriptor. The information concerning [the information about slipping] linear replacement MENTO to a primary defect list (Primary Defect List) is recorded on a secondary defect list (Secondary Defect List).

[0046] The media information descriptor (Media Information Descriptor) which is the field which records the information about media records the information about a zone etc. A drive information descriptor (Drive Information Descriptor) is a field which records the information about a drive (equipment which performs record reproduction of data to media). In a fixed drive, various information is recorded here.

[0047] An extended data descriptor (Extended Data Descriptor) is a physical bohrium information. A partition information and the extended information which was not able to be recorded into the logic bohrium information header are recorded.

[0048] Below, a bohrium data structure (Volume data structures) is explained. The size of a support descriptor (Anchor Descriptor) is below a physical sector size, and is recorded in the format shown in Table 4.

[0049]

[Table 4]

Anchor Descriptor

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Start Physical Sector Number of Main MIA	Uint32
12	4	Number of Physical Sectors in Main MIA	Uint32
16	4	Start Physical Sector Number of Reserve MIA	Uint32
20	4	Number of Physical Sectors in Reserve MIA	Uint32
24	2	Number of MIBs for MIA Map in Main MIA ($=x_1$)	Uint16
26	2	Number of MIBs for MIA Map in Reserve MIA ($=x_2$)	Uint16
28	$2x_1$	MIB Numbers of MIA Map for Main MIA in Main MIA	Uint16
$28+2x_1$	$2x_2$	MIB Numbers of MIA Map for Reserve MIA in Main MIA	Uint16
$28+2x_1+2x_2$	$2x_1$	MIB Numbers of MIA Map for Main MIA in Reserve MIA	Uint16
$28+4x_1+2x_2$	$2x_2$	Mib Numbers of MIA Map for Reserve MIA in Reserve MIA	Uint16

[0050] Signature (Signature:BP 0) As for the data-type field, 16 is set up. start physical SEKUTANAMBAOBU -- main -- MIA (Start Physical Sector Number of Main MIA:BP 8) is main -- the physical sector number of the physical sector of the head of MIA is specified NAMBAOBU physical SEKUTAIN main MIA (Number of Physical Sectors in Main MIA:BP 12) The number of the physical sectors of Main MIA is specified. Start physical SEKUTANAMBAOBU reserve MIA (Start Physical Sector Number of Reserve MIA:BP 16) The physical sector number of the physical sector of the head of Reserve MIA is specified. NAMBAOBU physical SEKUTAIN reserve MIA (Number of Physical Sectors in Reserve MIA:BP 20) The number of the physical sectors of Reserve MIA is specified. NAMBAOBU MIBs forehead MIA map in main MIA (Number of MIBs for MIA Map in Main MIA:BP 24) The size (the number of MIB) of the MIA map of Main MIA is specified. NAMBAOBU MIBs forehead MIA map in reserve (MIANumber of MIBs for MIA Map in Reserve MIA:BP 26) The

size (the number of MIB) of the MIA map of Reserve MIA is specified. MIB NAMBAOBU MIA map forehead main MIA in main MIA (MIB Numbers of MIAMap for Main MIA in Main MIA:BP 28) MIB in main MIA which is recording the MIA map to Main MIA is specified. The MIB number of MIB which constitutes a MIA map is set up in order.

[0051] MIB NAMBAOBU MIA map forehead reserve MIA in main MIA (MIB Numbers of MIA Map for Reserve MIA in Main MIA:BP 28+2x1) MIB in main MIA which is recording the MIA map to Reserve MIA is specified. The MIB number of MIB which constitutes a MIA map is set up in order. MIB NAMBAOBU MIA map forehead main MIA in reserve MIA (MIB Numbers of MIA Map for Main MIA in Reserve MIA : BP 28+2x1+2 x2) MIB under main reserve MIA which is recording the MIA map to MIA is specified. The MIB number of MIB which constitutes a MIA map is set up in order. MIB NAMBAOBU MIA map forehead reserve MIA in reserve MIA (MIB Numbers of MIA Map for Reserve MIA in Reserve MIA:BP 28+4x1+2 x2) MIB under reserve MIA which is recording the MIA map to Reserve MIA is specified. The MIB number of MIB which constitutes a MIA map is set up in order.

[0052] A MIA map (MIA Map) is used to show the operating condition of MIB. A MIA map shows the position of MIB which cannot be used according to MIB currently used for record of various kinds of data, a defect, etc., and intact MIB. A MIA map is recorded in the format shown in Table 5.

[0053]

[Table 5]

MIA Map

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Location of MIA Map	UInt16
10	2	Location of Volume Structure Descriptor	UInt16
12	2	Location of Media Information Descriptor	UInt16
14	2	Location of Drive Information Descriptor	UInt16
16	2	Location of Extended Data Descriptor	UInt16
18	2	Number of Map Entries (=n)	UInt16
20	2x1	Map Entries	bytes

[0054] Signature (Signature:BP 0) As for the data-type field, 2 is set up. ROKESHONOBU MIA map (Location of MIA Map:BP 8) The MIB number of the head MIB of a MIA map is specified. Location OBUBORYUMU structure descriptor (Location of Volume Structure Descriptor:BP 10) The MIB number of the head MIB of a bohrium structure descriptor is specified. Location OBUMEDIA information descriptor (Location of Media Information Descriptor:BP 12) The MIB number of the head MIB of a media information descriptor is specified. Location OBUDORAIBU information descriptor (Location of Drive Information Descriptor:BP 14) The MIB number of the head MIB of a drive information descriptor is specified.

[0055] Location OBUEKUSUTENDEDDO data descriptor (Location of Extended Data Descriptor:BP 16) The MIB number of the head MIB of an EKUSUTEN dead data descriptor is specified. Number OBUMAPPUENTORIZU (Number of Map Entries:BP 18) The number of entries of Map Entry which begins from BP 20 is specified. This number is equal to the number of MIB which exists in MIA, and is zero or less #FFF. Map en TORIZU (Map Entries:BP 20) The operating condition of MIB is specified. One Map Entry consists of UInt16, and, in the first MIB and the 2nd map entry, 2nd MIB, ..., the n-th map entry correspond [the first map entry] to n-th MIB. Table 6 is a table showing the value of map entry.

[0056]

[Table 6]

MIA Map Entry

Value	Interpretation
#0000-#FFEF	Next MIB Number
#FFF0	Unusable MIB
#FFF1	Unused MIB
#FFF2-#FFFE	Reserved
#FFFF	Last MIB of the data structure

[0057] Drawing 8 is drawing showing the structure of a bohrium structure descriptor (Volume Structure Descriptor). Here, @APS shows an aryne toe physical sector (Align to Physical Sector), and carrying out alignment of the data to a physical sector is shown. Moreover, as for the field from the next byte of the place where the data which should be recorded immediately before were actually recorded to the end of the sector, #00 are set up on the occasion of alignment.

[0058] A bohrium structure header (Volume Structure Descriptor Header) is recorded according to Table 7.

[0059]

[Table 7]

Volume Structure Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	UInt16
10	2	Reserved	#00 bytes
12	4	Offset to Physical Volume Information (=48)	UInt32
16	4	Offset to Partition Information (=416)	UInt32
20	4	Offset to Spare Area Information	UInt32
24	4	Offset to Logical Volume Information	UInt32
28	4	Offset to Defect List Information	UInt32

[0060] Signature (Signature:BP 0) As for the data-type field, 17 is set up. Descriptor size (Descriptor Size:BP 8) The size (the number of MIB) of a bohrium structure descriptor is specified. RIZABUDO (Reserved:BP 10) is reserved for a future extension, and assigns #00 to all bytes. OFUSETTOTU physical volume information (Offset to Physical Volume Information:RBP 12) The offset (byte count) from the head byte of the bohrium structure descriptor of a physical bohrium information is specified, and 48 is set up. An offset torpor TESHON information (Offset to Partition Information:RBP 16) specifies the offset (byte count) from the head byte of the bohrium structure descriptor of a partition information, and sets up 416. Offset spare area information (Offset to Spare Area Information:RBP 20) The offset (byte count) from the head byte of the bohrium structure descriptor of a spare area information is specified. An offset TUROJIKARU volume information (Offset to Logical Volume Information:RBP 24) specifies the offset (byte count) from the head byte of the bohrium structure descriptor of a logic bohrium information. OFUSETTOTU defect list information (Offset to Defect List Information:RBP 28) The offset (byte count) from the head byte of the bohrium structure descriptor of a defect list information is specified.

[0061] According to Table 8, you have to record a physical bohrium information (Physical Volume Information).

[0062]

[Table 8]

Physical Volume Information

RBP	Length	Name	Contents
0	2	Charactor Set	Charactor Set
2	2	Physical Volume Name Size	Uint16
4	256	Physical Volume Name	bytes
260	20	Physical Volume Identifier	bytes
280	6	Creation Time	Time Stamp
286	6	Modification Time	Time Stamp
292	2	Number of Partitions (=Np)	Uint16
294	2	Number of Spare Areas (=Ns)	Uint16
296	2	Number of Partitions with Defect Management (=Ndump)	Uint16
298	2	Number of Logical Volume (=Nv)	Uint16
300	2	Reserved	# 00 bytes
302	2	Extended Data Identifier	Uint16 bytes
304	64	Extended Data	bytes

[0063] Character sets (Charactor Set:RBP 0) The character code of the name of the physical bohrium recorded on the physical bohrium name field is specified. Physical volume name size (Physical Volume Name Size:RBP 2) The size (byte count) of the name of the physical bohrium recorded on the physical bohrium name field is specified. Physical volume name (Physical Volume Name:RBP 4) The name of physical bohrium is specified. Physical volume identifier (Physical Volume Identifier:RBP 260) The sequence of bytes for setting physical bohrium to a meaning practically is specified. A courier SHON time (Creation Time:RBP 280) specifies the time by which the volume structure of this physical bohrium was defined for the first time. A MODITIFIKESHON time (Modification Time:RBP 286) specifies the newest time by which the volume structure of this physical bohrium was changed. NAMBAOBU party SHON (Number of Partitions:RBP 292) specifies the number of the partitions contained in this physical bohrium, and is in agreement with the number of partition informations. [0064] Number OBUSU pair area (Number of Spare Areas:RBP 294) specifies the number of the spare area included in this physical bohrium, and is in agreement with the number of spare area informations. Number OBUPATESHONWIZU defect management (Number of Partitions with Defect Management:RBP 296) specifies the number of the partitions which perform defect management among the partitions contained in this physical bohrium, and is in agreement with the number of defect lists. Number OBUROJI cull volume (Number of Logical Volumes:RBP 298) specifies the number of the logic bohrium with which the partition contained in this physical bohrium belongs, and is in agreement with the number of logic bohrium informations. Reserve soil (Reserved:RBP 300) is reserved for a future extension, and assigns #00 to all bytes. EKUSUTEN dead data identifier (Extended Data Identifier:RBP 302) ID for specifying the EKUSU ten dead data currently recorded on the EKUSU ten dead data field and the EKUSU ten dead data area is specified. EKUSU ten dead data (Extended Data:RBP 304) are reserved for a future extension, and assign #00 to all bytes.

[0065] You have to record a partition information (Partition Information) in the format shown in Table 9.

[0066]

[Table 9]

Partition Information

RBP	Length	Name	Contents
0	4	Start Physical Sector Number	UInt32
4	4	Number of Physical Sectors	UInt16
8	4	Number of Usable Sectors	UInt16
12	4	Physical Sector size (=PSS)	UInt16
16	1	Access Type	UInt8
17	1	Usage Information	UInt8
18	2	Reserved	#00 bytes
20	4	Location of Primary Defect List	Desc Extent
24	4	Location of secondary Defect List	Desc Extent
28	2	Reserved	#00 bytes
30	2	Extended Data Identifier	UInt16
32	64	Extended Data	#00 bytes

[0067] Start physical sector number (Start Physical Sector Number:RBP 0) The physical sector number of the physical sector of the head of the field which constitutes a partition is specified. NAMBAOBU physical SEKUTAZU (Number of Physical Sectors:RBP 4) The number of the physical sectors of the field which constitutes a partition is specified. NAMBAOBU user bull SEKUTAZU (Number of Usable Sectors:RBP 8) The total of the physical sector which can be used among the fields which constitute a partition is specified, and it is in agreement with the number of the physical sectors of the field except the spare area included from all the fields of a partition to the partition field. Physical sector size (Physical Sector Size:RBP 12) The size (bytes number) of the physical sector of the field which constitutes a partition is specified. An access type (Access Type:RBP 16) specifies the state of the recording characteristic of this partition. Table 10 is a table showing the contents of an access type.

[0068]

[Table 10]

Access Type

Value	Name	Interpretation
0	Read Only	The user may not write any data in this partition
1	Write Once	The user can write data but once in this partition
2	Rewritable	The user can write data many times in this partition
3-15	Reserved	Reserved for futurer use

[0069] A YUZE gene formation (Usage Information:RBP 17) specifies the use state of this partition. Table 11 is a table showing the contents of a YUZE gene formation.

[0070]

[Table 11]

Usage information

Bit	Interpretation
0	Used (1 : used, 0 : not used)
1	Defect management : Slipping (1 : on, 0 : off)
2	Defect management : Linear replacement (1 : on, 0 : off)
3-7	Reserved

[0071] RIZABUDO (Reserved:RBP 18) is reserved for a future extension, and assigns #00 to all bytes. When the information about the position where the primary defect list was recorded on this field when a location OBUPURAIMARI defect list (Location of Primary Defect List:RBP 20) performed the defect management by slipping with this partition is stored and it does not perform the defect management by slipping, #00 are assigned to all bytes. When the information about the position where the secondary

defect list was recorded on this field when a location OBUSEKANDARI defect list (Location of Secondary Defect List:RBP 24) performed the defect management by linear replacement MENTO with this partition is stored and it does not perform the defect management by linear replacement MENTO, #00 are assigned to all bytes. RIZABUDO (Reserved:RBP 28) is reserved for a future extension, and assigns #00 to all bytes. EKUSUTEN dead data identifier (Extended Data Identifier:RBP 30) ID for specifying the EKUSU ten dead data currently recorded on the EKUSU ten dead data field and the EKUSU ten dead data area is specified. EKUSU ten dead data (Extended Data:RBP 32) are reserved for a future extension, and assign #00 to all bytes.

[0072] A spare area information (Spare Area Information) is recorded in the format shown in Table 12.

[0073]

[Table 12]

Spare Area Information

RBP	Length	Name	Contents
0	4	Start Physical Sector Number	Uint16
4	4	Number of Physical Sector	Uint16
8	8	Reserved	#00 bytes

[0074] A start physical sector number (Start Physical Sector Number:RBP 0) specifies the physical sector number of the physical sector of the head of spare area. NAMBAOBU physical sector (Number of Physical Sector:RBP 4) The number of the physical sectors which constitute a spare is specified. RIZABUDO (Reserved:RBP 8) is reserved for a future extension, and assigns #00 to all bytes.

[0075] A logic bohrium information header (Logical Volume Information Header) is recorded in the format shown in Table 13.

[0076]

[Table 13]

Logical Volume Information Header

RBP	Length	Name	Contents
0	2	Character Set	Character Set
2	2	Logical Volume Name Size	Unit16
4	256	Logical Volume Name	bytes
260	2	Boot Indicator	Unit16
262	2	File System Indicator	Unit16
264	2	Logical Sector Size	Unit16
266	2	Number of Partitions (=Npv)	Unit16
268	4	Reserved	#00 bytes
272	16	Logical Volume Contents Use	bytes
288	14	Reserved	#00 bytes
302	2	Extended Data Identifier	Unit16
304	64	Extended Data	#00 bytes

[0077] Character sets (Character Set:RBP 0) specify the character code of the name of the logic bohrium recorded on the logic bohrium name field. Logical volume name size (Logical Volume Name Size:RBP 2) specifies the size (byte count) of the name of the logic bohrium specified to be the logic bohrium name field. A logical volume name (Logical Volume Name:RBP 4) specifies the name of logic bohrium. A boot indicator (Boot Indicator:RBP 260) specifies the information about starting from logic bohrium. The contents of a boot indicator are shown in Table 14. A boot indicator is active and the logic bohrium which has the head partition in the physical bohrium must not be in [two or more] physical bohrium.

[0078]

[Table 14]

Boot Indicator

Value	Name	Contents
00h	Not Active	Physical volume is not set that computer boots up from this logical volume
80h	Active	Physical volume is set that computer boots up from this logical volume

[0079] A file system indicator (File System Indicator:RBP 262) specifies the file system currently used with this logic bohrium. The contents of a file system indicator are shown in Table 15.

[0080]

[Table 15]

File System Indicator

Value	Name	Contents
00h	Unknown	This logical volume is unknown.
01h	12bit FAT	This logical volume is formatted with 12bit FAT.
04h	16bit FAT	This logical volume is formatted with 16bit FAT.
05h	16bit FAT,Extended	This logical volume is formatted with 16bit FAT, and defined an extended partition.
06h	16bit FAT,Extended, 64KB/clster	This logical volume is formatted with 16bit FAT, and defined an extended partition, using 64KB/clster.
07h	HPFS	This logical volume is formatted with HPFS.
08h	32bit FAT	This logical volume is formatted with 32bit FAT.
F0h	KIFS	This logical volume is formatted with KIFS.

[0081] A logical sector size (Logical Sector Size:RBP 264) specifies the size (byte count) of the logical sector of this logic bohrium. NAMBAOBU party SHON (Number of Partitions:RBP 266) specifies the number of the partitions which constitute this logic bohrium, and is in agreement with the number of partition maps. RIZABUDO (Reserved:RBP 268) is reserved for a future extension, and assigns #00 to all bytes. A logical volume contents youth (Logical Volume Contents Use:RBP 272) is the field which the file system currently used with this logic bohrium may use freely. RIZABUDO (Reserved (RBP 288) is reserved for a future extension, and assigns #00 to all bytes.) EKUSUTEN dead data identifier (Extended Data Identifier:RBP 302) ID for specifying the EKUSU ten dead data currently recorded on the EKUSU ten dead data field and the EKUSU ten dead data area is specified. EKUSU ten dead data (Extended Data:RBP 304) are reserved for a future extension, and assign #00 to all bytes.

[0082] A partition map (Partition Map) is recorded in the format shown in Table 16.

[0083]

[Table 16]

Partition Map

RBP	Length	Name	Contents
0	20	Volume Identifier	bytes
20	2	Partition Number	Uint16
22	2	Reserved	#00 bytes

[0084] A volume identifier (Volume Identifier:RBP 0) specifies the physical bohrium identifier recorded on the physical bohrium information of the physical bohrium with which the partition which constitutes logic bohrium belongs. A party SHON number (Partition Number:RBP 20) specifies the partition number of the partition which constitutes logic bohrium. RIZABUDO (Reserved:RBP 22) is reserved for a future extension, and assigns #00 to all bytes.

[0085] A defect list information header (Defect List Information Header) is recorded in the format shown in Table 17.

[0086]

[Table 17]

Defect List Information Header

RBP	Length	Name	Contents
0	2	Number of MIB for Primary Defect List	Uint16
2	2	Number of MIB for Secondary Defect List	Uint16
4	12	Reserved	#00 bytes

[0087] The number of MIB currently used for a NAMBAOBU MIB forehead primary defect list (Number of MIB for Primary Defect List:RBP 0) recording a primary defect list is specified. The number of MIB currently used for a NAMBAOBU MIB forehead secondary defect list (Number of MIB for Secondary Defect List:RBP 2) recording a secondary defect list is specified. RIZABUDO (Reserved:RBP 4) is reserved for a future extension, and assigns #00 to all bytes.

[0088] Primary defect list / A secondary defect list (Primary Defect List/Secondary Defect List) is recorded in the format shown in Table 18.

[0089]

[Table 18]

Primary Defect List / Secondary Defect List

RBP	Length	Name	Contents
0	8	Signature	Signature
8	2	Partition Number	Uint16
10	2	Number of Entries (= Npd)	Uint16
12	4	Reserved	#00 bytes
16	4 (8) Npd	Defect List Entry	bytes

[0090] 19 is set up, when in a primary defect list 18 is set up and the data-type field of a signature (Signature:BP 0) is a secondary defect list. A party SHON number (Partition Number:BP 8) specifies the partition number of the partition which is using this defect list. Number OBUENTORIZU (Number of Entries:BP 10) specifies the number of entries of a defect list entry (Defect List Entry). RIZABUDO (Reserved:RBP 12) is reserved for a future extension, and assigns #00 to all bytes. In a primary defect list, a defect list entry (Defect List Entry:RBP 16) records a primary defect list entry, and, in a secondary defect list, records a secondary defect list entry. A defect list entry is recorded on the ascending order of the value of the physical SEKUTANAMBAOBU defect sector (Physical Sector Number of Defect Sector) field of each entry in both cases.

[0091] The structure of a media information descriptor (Media Information Descriptor) is shown in drawing 9.

[0092] A media information descriptor header (Media Information Descriptor Header) is recorded in the format shown in Table 19.

[0093]

[Table 19]

Media information Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	Uint16
10	6	Reserved	# 00 bytes
16	2	Number of discs	Uint16
18	2	Number of sides per disc	Uint16
20	2	Number of layers per side	Uint16
22	2	Number of zones per layer (=Nz)	Uint16
24	8	Reserved	# 00 bytes
32	2	Number of cylinders	Uint16
34	2	Number of heads (tracks per cylinder)	Uint16
36	2	Number of sectors per tracks	Uint16
38	10	Reserved	# 00 bytes

[0094] As for the data-type field of a signature (Signature:BP 0), 20 is set up. Descriptor size (Descriptor Size:BP 8) specifies the size (the number of MIB) of a media information descriptor. RIZABUDO (Reserved:BP 10) is reserved for a future extension, and assigns #00 to all bytes. Number OBUDISUKUSU (Number of discs:BP 16) specifies the number of disks. A number OBUSAIDAZU par disk (Number of sides per disc:BP 18) specifies the number of sides per disk. NAMBAOBU layer par side (Number of layers per side:BP 20) The number of layers per side is specified. A number OBUZONZU par layer (Number of zones per layer:BP 22) specifies the number of zones per layer. RIZABUDO (Reserved:BP 24) is reserved for a future extension, and assigns #00 to all bytes. Number OBUSHIRINDAZU (Number of cylinders:BP 32) specifies a number of cylinders. Number OBUHEZZU (Number of heads:(tracks per cylinder) BP34) specifies the number of heads (the number of trucks per cylinder). A NAMBAOBU sector part rack (Number of sectors per tracks:BP 36) specifies the number of sectors around a truck. RIZABUDO (Reserved:BP 38) is reserved for a future extension, and assigns #00 to all bytes.

[0095] A zone information (Zone Information) is recorded in the format shown in Table 20.

[0096]

[Table 20]

Zone Information

RBP	Length	Name	Contents
0	4	Start Physical Sector Number	Uint16
4	4	Number of Physical Sector	Uint16
8	8	Reserved	# 00 bytes

[0097] A start physical sector number (Start Physical Sector Number:RBP 0) specifies the physical sector number of the physical sector of the head of a zone. A NAMBAOBU physical sector (Number of Physical Sector:RBP 4) specifies the number of the physical sectors which constitute a zone. RIZABUDO (Reserved:RBP 8) is reserved for a future extension, and assigns #00 to all bytes.

[0098] The structure of a drive information descriptor (Drive Information Descriptor) is shown in drawing 10.

[0099] A drive information descriptor header (Drive Information Descriptor Header) is recorded in the format shown in Table 21.

[0100]

[Table 21]

Drive Information Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	Uint16
10	1	Strategy Type	Uint8
11	5	Reserved	# 00 bytes

[0101] As for the data-type field of a signature (Signature:BP 0), 21 is set up. Descriptor size (Descriptor Size:BP 8) specifies the size (the number of MIB) of a drive information descriptor. A strike radio-and-TV G type (Strategy Type:BP 10) specifies a strike radio-and-TV G type. RIZABUDO (Reserved:BP 11) is reserved for a future extension, and assigns #00 to all bytes.

[0102] The structure of an EKUSUTEN dead data descriptor (Extended Data Descriptor) is shown in drawing 11. Here, @APS shows an aryne toe physical sector (Align to Physical Sector), and it is shown that alignment of the data must be carried out to a physical sector. Moreover, as for the field from the next byte of the last data to the end of the sector, #00 are set up.

[0103] An EKUSUTENDEDDO data descriptor header (Extended Data Descriptor Header) is recorded in the format shown in Table 22.

[0104]

[Table 22]

Extended Descriptor Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Descriptor Size	Uint16
10	6	Reserved	# 00 bytes
16	2	Location of Extended Data for Physical Volume	Desc Extent
20	4Np	Location of Extended Data for Partitions	Desc Extent
20+4Np	4Nv	Location of Extended Data for Logical Volume	Desc Extent

[0105] As for the data-type field of a signature (Signature:BP 0), 22 is set up. Descriptor size (Descriptor Size:BP 8) specifies the size (the number of MIB) of an EKUSUTEN dead data descriptor. RIZABUDO (Reserved:BP 10) is reserved for a future extension, and assigns #00 to all bytes. Location OBUEKUSUTENDETTODETAFOA physical volume (Location of Extended Data for Physical Volume:BP 16) specifies the place where the extended data about this physical bohrium are recorded. Location OBUEKUSUTENDEDDODETAFOA party SHON (Location of Extended Data for Partitions:BP 20) specifies the place where the extended data about each partition are recorded. Location OBUEKUSUTENDEDDODETAFOA logical volume (20+4Np of Location of Extended Datafor Logical Volume:BP) specifies the place where the extended data about each logic bohrium are recorded.

[0106] Below, the level (Levels of medium interchange) of media exchange is described. The level 1 of media exchange prepares the following limits. That is, logic bohrium consists of partitions belonging to the same physical bohrium. When two or more partitions are defined as the same physical bohrium, the field of a partition must not lap. All the physical sectors of the partition which constitutes logic bohrium have the same physical sector size. A logic sector size is the multiple of a physical sector size, or a physical sector size is the multiple of a logic sector size. The size of a partition is the multiple of the value of the one where a logic sector size or a physical sector size is larger. The partition which performs defect management surely secures one or more spare area. The defect management by linear replacement MENTO uses the spare area secured in the partition as an alternative data area.

[0107] The level 1 of media exchange does not have a limit.

[0108] Below, the example (Example of volume structure) of a bohrium structure is explained. Table 23 is a table showing the example of the volume structure of FAT in VDR, ISO9660 (with Joliet), ISO/IEC13346, and the hybrid disk of KIFS. It is shown that ◇ of Table 23 is the position constant which cannot be rearranged.

[0109]

[Table 23]

Example of volume structure (FAT, 9660, 13346, KIFS Hybrid)

PSN(hax)	Descriptor	Contents
0	[FAT] Partition Table	◆[FAT] Partition Table
-	-	-
c	[KIFS] Anchor Descriptor	◆[KIFS] Anchor
-	-	-
10	[9660] Primary Volume Descriptor	◆[9660/13346/KIFS] Volume recognition Sequence
11	[9660] Primary Volume Descriptor (Reserve)	
12	[9660] Supplementary Volume Descriptor (for Joliet)	
13	[9660] Volume Descriptor Set Terminator	
14	[13346] Beginning Extended Area Descriptor	
15	[13346] NSR Descriptor	
16	[13346] Terminating Extended Area Descriptor	
17	[13346] Beginning Extended Area Descriptor	
18	[KIFS] KIFS Descriptor	
19	[13346] Terminating Extended Area Descriptor	
-	-	-
30	[13346] Primary Volume Descriptor	[13346] Main Volume Descriptor Sequence Extent
31	[13346] Implementation Use Volume descriptor	
32	[13346] Partition Descriptor	
33	[13346] Logical Volume Descriptor	
34	[13346] Unallocated Space Descriptor	
35	[13346] Terminating Descriptor	
-	-	-
40	[13346] Primary Volume Descriptor	[13346] Reserve Volume Descriptor sequence Extent
41	[13346] Implementation Use Volume Descriptor	
42	[13346] Partition Descriptor	
43	[13346] Logical Volume Descriptor	
44	[13346] Unallocated Space Descriptor	
-	-	-
80	[KIFS] MIA Map for Main MIA	[KIFS] Main MIA
81	[KIFS] MIA Map for Reserved MIA	
82	[KIFS] Volum Structure Descriptor	
83	[KIFS] Primary Defect List	
84	[KIFS] Secondary Defect List	
85	[KIFS] Media Information Descriptor	
86	[KIFS] Drive Information Descriptor	
87	[KIFS] Extended Data Descriptor	
88	[KIFS] Extended Data	
-	-	
c0	[KIFS] MIA Map for Reserve MIA	[KIFS] Reserve MIA
c1	[KIFS] MIA Map for Main MIA	
c2	[KIFS] Volum Structure Descriptor	
c3	[KIFS] Primary Defect List	
c4	[KIFS] Secondary Defect List	
c5	[KIFS] Defect Sector	
c6	[KIFS] Media Information Descriptor	
c7	[KIFS] Drive Information Descriptor	
c8	[KIFS] Extended Data Descriptor	
c9	[KIFS] Extended Data	
-	-	-
100	[13346] Anchor Volume Descriptor Pointer	◆[13346] Anchor
-	-	-
180	[KIFS] LOGICAL VOLUME	-
-	-	-
LP8N-100	-	-
-	-	-
LP8N-180	[13346] Anchor Volume Descriptor Pointer	◆[13346] Anchor
-	-	-
LP8N-20	[KIFS] Anchor Descriptor	◆[KIFS] Anchor
-	-	-
LP8N-c	[KIFS] Anchor Descriptor	◆[KIFS] Anchor
-	-	-
LP8N	[13346] Anchor Volume Descriptor Pointer	◆[13346] Anchor

[0110] Next, AV file system (AV File System) constituted on logic bohrium is explained. A logical sector number (Logical Sector Number) is a number given in order to discriminate a logical sector.

Logic bohrium (Logical Volume) is the set which was by carrying out with the logical sector number which begins from 0 of ascending order continuously etc., and consisted of logical sectors of a size. .
 [0111] File system administrative management information area (Management Information Area (MIA)) is a field which consists of two or more logical sectors followed on the logic bohrium which stores various kinds of control information of AV file system. A management information block (Management Information Block (MIB)) is a logical sector in MIA. A management information block number (Management Information Block Number (MIB number)) is a logical sector number of a management information block. It has the value which lengthened the logical sector number of the head management information block of MIA from Number.

[0112] Below, the whole AV file system is explained. AV file system descriptor (AV File System Descriptor) mentioned later is recorded in one logical sector, and specifies the position of the main MIA on logic bohrium, the position of Reserve MIA, a size, and the MIA map on main MIA and Reserve MIA. The position of AV file system descriptor is set up as shown in Table 24 in the logical volume contents youth (Logical Volume Contents Use:BP 284) field of the above-mentioned logic bohrium information header.

[0113]

[Table 24]

Logical Volume Contents Use field

RBP	Length	Name	Contents
0	4	Main AV File System Descriptor Location	Uint32
4	4	Reserve AV File System Descriptor Location	Uint32
8	8	Reserved	#00 bytes

[0114] A main AV file system descriptor location (Main AV File SystemDescriptor Location:RBP 0) specifies the logical sector number of AV file system descriptor. A reserve AV file system descriptor location (Reserve AV File System Descriptor Location:RBP 4) specifies the logical sector number of AV file system descriptor in a place different from having been specified on the main AV file system descriptor location. #FFFFFFFF is set to a reserve AV file system descriptor location when AV file system descriptor exists on logic bohrium only in one piece. RIZABUDO (Reserved:RBP 8) is reserved for extension and #00 are set up.

[0115] Various kinds of management information of AV file system is file system administrative management information area. (Management Information Area:MIA) It is recorded. For reliability reservation, MIA with the management information of equal contents is recorded on two on logic bohrium, and calls main MIA and Reserve MIA, respectively. The position of the position of Main MIA and Reserve MIA, a size, and the MIA map in MIA is prescribed by AV file system descriptor. The logical sector in MIA is a management information block. (MIB) It is called and the offset from the head MIB of MIA of the logical sector number is a management information block number. (MIB number) It is called.

[0116] As for specification of MIB, an MIB number is used. MIB which cannot use MIA according to a defect etc., intact MIB, And the MIA map (MIA Map), file table (File Table) which are a data structure, Allocation extent table (Allocation Extents Table) An allocation strike RATEJII table (Allocation Strategy Table), A defect information table (Defect Information Table (Optional)), And it consists of MIB used to store an EKUSUTEN dead attribute table (Extended Attribute Table (Optional)). It is recorded on a MIA map for which purpose MIB in MIA is used. Various kinds of data structures are stored in the inside of one MIB, or two or more MIB. It is Map in a MIA map which MIB when a data structure is recorded on two or more MIB, is connected in which turn. It is recorded on the entry field. As the data structure was MIB, when it finishes, #00 are stored from the next byte of the end of data to the byte of the last of the MIB.

[0117] In AV file system, a file and a directory are managed on the file table mentioned later. The

structure of a file table is determined by the file table structure type (File Table Structure Type) which is a parameter in a file table header. In the file table structure type 0, a file table consists of a file table header and one or more file records. The field for a file record being the data area of a fixed length, and discriminating a file record, The field and creation showing the kind of file record, and the field showing the time of a modification date, The field showing the position and size of data, the field showing an attribute, the field that points out the parent file record called a manager of a youth hostel link (Parent Link), The field which points out the brother file record called the NeXT link (Next Link), It consists of the field which points out the child file record called a tea yield link (Child Link), and the field which points out EKUSUTENDEDDO attribute record CHIEIN (Extended Attribute Record Chain). The number by which a file record is called a file record number is attached, and a manager of a youth hostel link, the NeXT link, and a tea yield link are specified using this file record number.

[0118] By the file table structure type 0, the tree structure as shown in drawing 12 from which the file record of the beginning of a file table serves as the root is built. The circle in drawing expresses one file record, and the file record of the root is called a root file record (Root File Record). The file record without the data which should be referred to is called a directory, and the file record with data is called a file. Not only a directory but a file can have a child file record. This layered structure is realized by setting up a tea yield link (Child Link), the NeXT link (Next Link), and a manager of a youth hostel link (Parent Link), as shown in drawing 13.

[0119] The list of file records which consist of neck strike links is called file record chain, and there must not be two or more records which are the same files ID and have the same file type during this list. A subfile is a kind of a file, and it shows some data which a parent file record refers to as if it was another file. The file record with which 10 was set as the data location type (Data Location Type) of the attribute (Attribute) field expresses a subfile.

[0120] In AV file system, management of the data which made the unit the field where it continued on logic bohrium called an allocation extent (Allocation Extent) is performed. An allocation extent is ended by arbitrary byte offset of the logical sector following it including zero or more logical sectors which begin from arbitrary byte offset of a logical sector, and are completed by the arbitrary byte offset in the logical sector, or continue. The start point of the allocation extent t, an ending point, an attribute, etc. are recorded on the allocation extent record in an allocation extent table.

[0121] The allocation extent record corresponding to all the allocation extents on logic bohrium is registered into an allocation extent table. An allocation extent record has the field indicating the following allocation extent record, and can create the list which consists of two or more allocation extent records using this field. This list is called an allocation extent record chain. Usually, file data is treated as a set with sequence of the allocation extent corresponding to an allocation extent record chain.

[0122] The list made from the allocation extent record (the allocation extent record status is the record of 00) with which it is not used in the allocation extent table is called free allocation extent REKODOCHIEIN, and can be easily followed from an allocation extent table. Moreover, allocation extent record judged as a problem being in reuse including a defective (defect) sector in a corresponding allocation extent (record which has 10 for the allocation extent record status) The list collected and created can be called a DIFEKUTIBUAROKESHON extent record chain, and this list can also be easily followed from an allocation extent table.

[0123] It is determined by allocation strike radio-and-TV G (Allocation Strategy) on which position of logic bohrium an allocation extent is put. An allocation strike RATEJII table can register two or more allocation strike radio-and-TV G, and different allocation strike radio-and-TV G for every file can be used for it, and it can arrange an allocation extent on logic bohrium. The parameter which the range of the field which each allocation strike radio-and-TV G manages, or allocation strike radio-and-TV G uses is recorded on the allocation strike RATEJII record in an allocation strike RATEJII table. In the file table structure type 0, allocation strike radio-and-TV G is determined for every file record, and is recorded on the data location field of a file record. This data location field is referred to in the case of operation of an allocation extent, and corresponding allocation strike radio-and-TV G is called.

[0124] Two allocation strike RATEJII types, the allocation strike RATEJII type 0 (Allocation Strategy

Type 0) and the allocation strike RATEJII type 1 (Allocation Strategy Type 1), are defined. The allocation strike RATEJII type 0 is the method for which it was suitable when the file of the comparatively small size of index data etc. was dealt with in discontinuous, and the allocation strike RATEJII type 1 is a method suitable for writing data continuously [MPEG etc.].

[0125] A defect information table (Defect Information Table) is a table which recorded the logical sector number of the defective sector in logic bohrium, and is applicable to management of a defective sector.

[0126] An EKUSUTEN dead attribute table (Extended Attribute Table) can be used in order to hold a file or the extended attribute of a directory in MIA. An EKUSUTEN dead attribute table consists of an EKUSUTENDEDDO attribute table header and one or more EKUSUTENDEDDO attribute table records. An EKUSUTEN dead attribute record is a record of the fixed length which has the field for a link, and can create the EKUSUTENDEDDO attribute record chain which considered two or more EKUSUTEN dead attribute records as the list.

[0127] As for the head of the data structure which AV file system uses, a signature (Signature) is set up. A signature is recorded as shown in Table 25.

[0128]

[Table 25]

Signature

RBP	Length	Name	Contents
0	4	Identification	bytes = " AVFS"
4	1	Version	UInt8 = 1
5	1	Data type	UInt8
6	2	Reserved	#00 bytes

[0129] As for identification (Identification:RBP 0), character string "AVFS" is set up according to ISO/IEC 646. A version (Version:RBP 4) specifies a version number and 1 is set up. A data type (Data type:RBP 5) specifies the kind of data structure. The value shown in Table 26 is set up according to the kind of data structure.

[0130]

[Table 26]

Data type

Value	Interpretation
0	Reserved
1	AV File System Descriptor
2	MIA Map
3	File Table
4	Allocation Extents Table
5	Allocation Strategy Table
6	Defect Information Table
7	Extended Attribute Table
8-255	Reserved

[0131] RIZABUDO (Reserved:RBP 6) is reserved for extension and #00 are set up. A signature is used in order to discriminate a data structure at the time of crash recovery.

[0132] AV file system descriptor (AV File System Descriptor) is recorded as shown in Table 27.

[0133]

[Table 27]

AV File System Descriptor

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Location of Main MIA	UInt32
12	4	Location of Reserve MIA	UInt32
16	2	Length of Main MIA	UInt16
18	2	Length of Reserve MIA	UInt16
20	4	Creation Time	Time Stamp
24	4	Modification Time	Time Stamp
28	2	Number of MIA Map Sectors in Main MIA (=x ₁)	UInt16
30	2	Number of MIA Map Sectors in Reserve MIA (=x ₂)	UInt16
32	2x ₁	MIA Map Sectors in Main MIA	bytes
32+2x ₁	2x ₂	MIA Map Sectors in Reserve MIA	bytes

[0134] As for the data-type field of a signature, 1 is set up. location OBUMEIN MIA (Location of Main MIA:BP 8) is main -- the start logical sector number of MIA is specified Location OBURIZABU MIA (Location of Reserve MIA:BP 12) specifies the start logical sector number of Reserve MIA. length OBUMEIN MIA (Length of Main MIA:BP 16) is main -- the size of MIA is specified with the number of logical sectors The RENGUSUOBU reserve MIA (Length of Reserve MIA:BP 18) specifies the size of Reserve MIA with the number of logical sectors. A courier SHON time (Creation Time:BP 20) stores the time which created AV file system descriptor. A MODITIFIKESHON time (Modification Time:BP 24) specifies the time which updated AV file system descriptor. The NAMBAOBU MIA map sector in main MIA (Number of MIA Map Sectors in Main MIA:BP 28) specifies the number of the MIB numbers described by main MIA map SEKUTAZU (Main MIA Map Sectors:BP 32).

[0135] The NAMBAOBU MIA map sector ZUIN reserve MIA (Number of MIA Map Sectors in Reserve MIA:BP 30) specifies the number of the MIB numbers described by reserve MIA map SEKUTAZU (Reserve MIA Map Sectors:BP 32+2x₁). The MIA map sector in main MIA (MIA Map Sectors in Main MIA:BP 32) specifies MIB which constitutes the MIA map in main MIA, and the MIB number of MIB which constitutes a MIA map is set up in order. The MIA map sector ZUIN reserve MIA (MIA Map Sectors in Reserve MIA:BP 32+2x₁) specifies MIB which constitutes the MIA map under reserve MIA, and the MIB number of MIB which constitutes a MIA map is set up in order.

[0136] A MIA map (MIA Map) is used for the operating condition of MIB in MIA being shown. A MIA map shows the position of MIB and intact MIB which cannot be used according to various kinds of data structures in MIA, a defect, etc. A MIA map is recorded as shown in Table 28.

[0137]

[Table 28]

MIA Map

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Location of MIA Map	UInt16
10	2	Location of Allocation Strategy Table	UInt16
12	2	Location of File Table	UInt16
14	2	Location of Allocation Extents Table	UInt16
16	2	Location of Defect List Table	UInt16
18	2	Location of Extended Attribute Descriptor	UInt16
20	2	Reserved	bytes
22	2	Number of Map Entries (=x ₁)	UInt16
24	2x ₁	Map Entries	bytes

[0138] As for the data-type field of a signature (Signature:BP 0), 2 is set up. A ROKESHONOBU MIA

map (Location of MIA Map:BP 8) specifies the MIB number of the head MIB of the MIA map in this MIA. Location OBUA location SUTORATEJIITEBURU (Location of Allocation Strategy Table:BP 10) specifies the MIB number of the head MIB of the allocation strike RATEJII table in this MIA. A location file table (Location of File Table:BP 12) specifies the MIB number of the head MIB of the file table in this MIA.

[0139] A location OBUAROKESHONEKUSU tense table (Location of Allocation Extents Table:BP 14) specifies the MIB number of the head MIB of the allocation extent table in this MIA. A ROKESHONOBU defect list table (Location of Defect List Table:BP 16) specifies the MIB number of the head MIB of the defect list table in this MIA. #FFFF is set when a defect list table does not exist in this MIA. A location OBUEKUSUTENDEDDO attribute descriptor (Location of Extended Attribute Descriptor:BP 18) specifies the MIB number of the head MIB of the EKUSUTEN dead attribute descriptor in this MIA. #FFFF is set when an EKUSUTEN dead attribute descriptor does not exist in this MIA. RIZABUDO (Reserved:BP 20) is reserved for extension and #00 are set up.

[0140] NAMBAOBU map en TORIZU (Number of Map Entries:BP 22) specifies the number of entries of the map entry which begins from (BP 24). This number is equal to the number of MIB which exists in MIA, and is zero or less #FFF. Map en TORIZU (Map Entries:BP 24) specifies the operating condition of MIB in this MIA. one map entry -- from Uint16 -- becoming -- the first map entry -- MIB of the beginning of MIA, and the 2nd map entry -- 2nd MIB -- it corresponds to ...

[0141] The value of map entry has the meaning shown in Table 29.

[0142]

[Table 29]

Map entry value

Value	Interpretation
#0000-#FFEF	Next MIB Number
#FFF0	Unusable MIB
#FFF1	Unused MIB
#FFF2-#FFFE	Reserved
#FFFF	Last MIB of the data structure

[0143] A data structure is equal to a logic sector size, or small, and #FFFF is set to the map entry corresponding to the MIB when stored in one MIB. #FFFF is set to the map entry corresponding to MIB other than the last by the map entry corresponding to the last MIB in the MIB number of the following MIB when a data structure is recorded over two or more MIB. MIB whose value of map entry is #FFF1 shows that the block is intact, and when a data structure needs new MIB, it can use it. MIB whose value of map entry is #FFF0 expresses the things (defective sector etc.) which a problem is in the use.

[0144] A file table (File Table) consists of a file table header and file table data, as shown in drawing 14. The structure of file table data is decided by the File Table Structure Type field of a file table header.

[0145] A file table header (File Table Header) is recorded as shown in Table 30.

[0146]

[Table 30]

File Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Length of File Table Data	Uint32
12	2	File Table Structure Type	Uint16
14	18	File Table Structure Type dependent information	bytes

[0147] As for the data-type field of a signature (Signature:BP 0), 3 is set up. RENGUSUOBU file table

data (Length of File Table Data:BP 8) specify the length of file table data by the byte count. A file table structure type (File Table Structure Type:BP 12) specifies the structure of file table data. The information the file table SUTORAKUCHATAIPUDIPENDENTO information (File Table Structure Type dependent information:BP 14) was decided to be for every file table structure type is set up.

[0148] When a file table structure type (File Table Structure Type) is 0, a file table consists of a file table header and one or more file records, as shown in drawing 15 . The number of the continuation in which a file record begins from 0, and ascending order is attached, and this number is called a file record number. The list of file records is made by setting the file record number of a next record as the NeXT link (Next Link) field, and this list is called a file record chain. All the file records with which it is not used in a file table create the file record chain called a free file record chain.

[0149] When a file table structure type is 0, a file table header (File Table Header) must be recorded as shown in Table 31.

[0150]

[Table 31]

File Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Length of File Table Data	UInt32
12	2	File Table Structure Type	UInt16 (=0)
14	2	Number of File Records	UInt16
16	2	First Free File Record	UInt16
18	14	Reserved	# 00bytes

[0151] As for the data-type field of a signature (Signature:BP 0), 3 is set up. The number with which RENGUSUOBU file table data (Length of File Table Data:BP 8) covered NAMBAOBU file records (Number of File Records:BP 14) over the length of a file record is set up. As for a file table structure type (File Table Structure Type:BP 12), 0 is set up. NAMBAOBU file REKODOZU (Number of File Records:BP 14) specifies the number of file records which constitutes a file table. The number of file records takes an one or more # zero or less FFF value. #FFFF is set up, when first free file records (First Free File Records:BP 14) point out the element of the beginning of a free file record chain and a free file record does not exist in a file table. RIZABUDO (Reserved:BP18) is reserved for extension and #00 are set up.

[0152] A file record (File Record) must be recorded as shown in Table 32.

[0153]

[Table 32]

File Record of File Structure Type 0

RBP	Length	Name	Contents
0	2	File ID	UInt16
2	2	File Type	UInt16
4	4	Attribute	UInt32
8	4	Creation Time	Time Stamp
12	4	Modification Time	Time Stamp
16	8	Data Length	UInt64
24	8	Data Location	bytes
32	2	Child Link	UInt16
34	2	Next Link	UInt16
36	2	Parent Link	UInt16
38	2	Extended Attribute Record Number	UInt16

[0154] File ID (File ID:RBP 0) specifies the number for discriminating a file record with the same file type under file record chain. A file type (File Type:RBP 2) specifies the number for directing the kind of

this file record. An attribute (Attribute:RBP 4) specifies the attribute of the data which this file record or this file record refers to. A courier SHON time (Creation Time:RBP 8) specifies the date and time of creation of this file record. A modification time (Modification Time:RBP 12) specifies the change time of the data which this file record or a file record refers to. Data length (Data Length:RBP 16) specifies from a byte the length of the data which a data location (Data Location:RBP 24) refers to, and when there are no data to refer to, it sets 0. A data location (Data Location:RBP 24) specifies the position of the data which this file record refers to. The interpretation of the field changes with the data location type (Data Location Type:Bit 1-2) contents of an attribute (Attribute:RBP 4). #FFFF is set up, when a tea yield link (Child Link:RBP 32) specifies the file record number of the Child file record and such a file record does not exist. The NeXT link (Next Link:RBP 34) specifies the file record number of the following file record which constitutes a file record chain, and #FFFF is set up when this file record is the element of the last of a file record chain.

[0155] A manager of a youth hostel link (Parent Link:RBP 36) specifies the file record number of a manager of a youth hostel file record, and when this file record is a root file record, its own file record number, 0 [i.e.,], is set up. #FFFF is set up, when an EKUSUTENDEDDO attribute record number (Extended Attribute Record Number:RBP 38) specifies the EKUSUTEN dead attribute record number of the head of the EKUSUTENDEDDO attribute record chain which this file record uses and it does not refer to an EKUSUTEN dead attribute record.

[0156] The attribute (Attribute) field is recorded as shown in Table 33.

[0157]

[Table 33]

Attribute of the File Record

Bit	Interpretation
0	Valid
1-2	Data Location Type
3	Protected
4	Sorted
5-31	Reserved

[0158] BARIDDO (Valid:Bit 0) expresses whether this file record is an effective record, and, in the case of 0, means that this file record is not used, and a file record has it during a free file record chain. In the case of 1, BARIDDO means that this file record is used, and it is a root file record to a tea yield link. It can reach through a neck strike link. A data location type (Data Location Type:Bit 0-1) specifies the format of a data location (Data Location:RBP 24). When a data location type is 00, it is shown that a data location does not have what is referred to (this value is set when a file record is a directory). A data location is expressed with the format which shows the allocation extent record number of the head of an allocation extent record chain, and an allocation strike radio-and-TV G number in Table 34 when a data location type is 01. When a data location type is 10, it expresses that a file record is a subfile and the offset from the head of the data with which the data location of a manager of a youth hostel file record expresses a data location is expressed with Uint64. The data location type of 11 is reserved for extension.

[0159]

[Table 34]

Data Location file of Type 01

RBP	Length	Name	Contents
0	2	Reserved	#00 bytes
2	2	Allocation Strategy Number	Uint16
4	4	First Allocation Extent Record Number	Uint32

[0160] Pro TEKUTEDDO (Protected:Bit 3) means that this file record is protected. It means that the file record chain to which this file record belongs is sorted by order with a young file type, and sow TEDDO (Sorted:Bit 4) is sorted by the young order of File ID in the still more nearly same file type.

RIZABUDO (Reserved:Bit 5-31) is reserved for extension.

[0161] An allocation extent table (Structure of the Allocation Extents Table) consists of an allocation extent table header and an allocation extent record, as shown in drawing 16 . The number of the continuation which begins from 0, and ascending order is given to an allocation extent record. This number is called an allocation extent record number. The list of allocation extent records is built by setting the allocation extent record number of a next record as NeXT allocation extent REKODOFIRUDO. This list is called an allocation extent record chain.

[0162] An allocation extent table header (Allocation Extents Table Header) is recorded as shown in Table 35.

[0163]

[Table 35]

Allocation Extents Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Number of Allocation Extent Records	UInt32
12	4	First Free Allocation Extent Record	UInt32
16	4	First Defective Allocation Extent	UInt32
20	4	Reserved	#00 bytes

[0164] As for the data-type field of a signature (Signature:BP 0), 4 is set up. NAMBAOBU allocation extent records (Number of Allocation ExtentRecords (BP 8) specifies the number of the allocation extent records in an allocation extent table.) A FASUTOFURI allocation extent record (First Free Allocation Extent Record:BP 12) points out the element of the beginning of free allocation extent REKODOCHIEIN.

[0165] When a free allocation extent record does not exist in an allocation extent table, #FFFFFFFF is set as this field. A FASUTODIFEKUTIBU allocation extent record (First DefectiveAllocation Extent Record:BP 16) points out the element of the beginning of a DIFEKUTIBUAROKESHON extent record chain. When a DIFEKUTIBU allocation extent record does not exist in an allocation extent table, #FFFFFFFF is set as this field. RIZABUDO (Reserved:BP 20) is reserved for extension and #00 are set up.

[0166] An allocation extent record (Allocation Extent Record) expresses the position of the starting position of an allocation extent, an end position, an attribute, and the following allocation extent record that constitutes an allocation extent record chain. An allocation extent record is recorded as shown in Table 36.

[0167]

[Table 36]

Allocation Extent Record

RBP	Length	Name	Contents
0	4	Start Logical Sector Number	Uint32
4	1	Allocation Strategy Number	Uint8
5	1	Reserved	Uint8
6	2	Start Offset	Uint16
8	4	End Logical Sector Number	Uint32
12	2	Reserved	Uint16
14	2	End Offset	Uint16
16	4	Attribute	Uint32
20	4	Next Allocation Extent Record	Uint32
24	8	Length of the Allocation Extent	Uint64

[0168] A start logical sector number (Start Logical Sector Number:RBP 0) specifies the logical sector containing the start byte of an allocation extent, and a logical sector number is set up. An allocation strike RATEJII number (Allocation Strategy Number:RBP 4) directs according to which allocation strike radio-and-TV G this allocation extent record is arranged. RIZABUDO (Reserved:RBP 5) is reserved for extension and #00 are set up. Start offset (StartOffset:RBP 6) specifies the byte offset from the head byte of the logical sector containing the start byte of an allocation extent to a start byte, and 0 will be set if a starting position is equal to the head byte of the logical sector.

[0169] And a logical sector number (End Logical Sector Number:RBP 8) specifies the logical sector number of the logical sector containing the last byte of an allocation extent. RIZABUDO (Reserved:RBP 12) is reserved for extension and #00 are set up. And offset (End Offset:RBP 14) specifies the offset from the head byte of a logical sector including the end byte of an allocation extent to an end byte, and 0 will be set if an end byte is equal to the head byte of the logical sector. The value which an attribute (Attribute:RBP 16) expresses has the meaning shown in Table 37.

[0170]

[Table 37]

Attribute of the Allocation Extent Record

Bit	Interpretation
0-1	Allocation Extnt Record Status
1-31	Reserved

[0171] Allocation extent record status (Bit 0-1) When it is 01, this allocation extent record points out an effective allocation extent, and can do read-out normally. When this bit is 11, this allocation extent record has pointed out the effective allocation extent, and means that read-out may be normally impossible by existence of a defective sector etc. It means that it can be used in case this allocation extent record is not used now but a new allocation extent is arranged, when this bit is 00. When this bit is 10, using it, in order that the allocation extent which this allocation extent record points out may arrange a new allocation extent since the defective sector is included, although somewhere are not referred to, either expresses that it is not suitable. RIZABUDO (Reserved:Bit 2-31) is reserved for extension, and 0 is set up.

[0172] The NeXT allocation extent record (Next Allocation Extent Record:RBP 20) specifies the following allocation extent record number which constitutes an allocation extent record chain. #FFFFFFFF is set when an allocation extent record is the element of the last of an allocation extent record chain. A length OBUZA allocation extent (Length of the Allocation Extent:RBP 24) directs the length of the allocation extent which this allocation extent record directs by the byte count. a start logical sector number (Start Logical Sector Number:RBP 0), start offset (Start Offset:RBP 6), and a logical sector number (End Logical Sector Number:RBP 8) -- and -- and the byte count called for by calculation from offset (End Offset:RBP 14) and the byte count set to this field are equal

[0173] An allocation strike RATEJII table specifies all the allocation strike radio-and-TV G currently used for A/V file system arranging data with this logic bohrium. An allocation strike RATEJII table consists of an allocation SUTORATEJII table header and an allocation strike RATEJII record, as shown in drawing 17 .

[0174] An allocation SUTORATEJII table header (Allocation Strategy Table Header) is recorded as shown in Table 38.

[0175]

[Table 38]

Allocation Strategy Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Number of Allocation Strategy Record	Uint16
10	6	Reserved	#00 bytes

[0176] As for the data-type field of a signature (Signature:BP 0), 5 is set up. A NAMBAOBU allocation SUTORATEJII record (Number of Allocation Strategy Record:BP 8) is allocation strike radio-and-TV G. The number of the allocation strike RATEJII records in a table is specified. RIZABUDO (Reserved:RBP 10) is reserved for extension and #00 are set up.

[0177] An allocation strike RATEJII record is used for specifying allocation strike radio-and-TV G. An allocation strike RATEJII record is recorded as shown in Table 39.

[0178]

[Table 39]

Allocation Strategy Record

RBP	Length	Name	Contents
0	2	Length of Allocation Strategy Record	Uint16
2	2	Allocation Strategy Type	Uint16
4	1	Allocation Strategy Number	Uint8
5	3	Reserved	#00 bytes
8	x1	Allocation Strategy Type Dependent Data	bytes

[0179] A RENGUSUOBU allocation SUTORATEJII record (Length of Allocation Strategy Record:RBP 0) specifies the length of this allocation strike RATEJII record by the byte count, and the length is the multiple of 8. An allocation strike RATEJII type (Allocation Strategy Type:RBP 2) specifies the kind of this allocation strike RATEJII record. An allocation strike RATEJII number (Allocation Strategy Number:RBP 4) specifies a record of what position in an allocation strike RATEJII table this allocation strike RATEJII record is, and 0 will be set if this record is the first record. RIZABUDO (Reserved:RBP 5) is reserved for extension and #00 must be set up. The contents it was decided for every allocation strike RATEJII type that allocation SUTORATEJIIITAIPUDIPEN dent data (Allocation Strategy Type Dependent Data:RBP 8) would be are set.

[0180] The following conditions are satisfied in the allocation strike RATEJII type 0. the 1st -- an allocation extent -- the start logical sector number (Start Logical Sector Number:RBP 8) of an allocation strike RATEJII record -- and -- and it must be arranged in the field specified by the logical sector number (End Logical Sector Number:RBP 12) When a part of logical sector is assigned [2nd] to a certain allocation extent, no byte of the logical sector belongs to another allocation extent. The head of an allocation extent and the head of a logical sector are in agreement with the 3rd. Allocation strike RATEJII type The allocation strike RATEJII record of 0 is recorded as shown in Table 40.

[0181]

[Table 40]

Allocation Strategy Record of Allocation Strategy Type 0

RBP	Length	Name	Contents
0	2	Length of Allocation Strategy Record	UInt16 (= 16)
2	2	Allocation Strategy Type	UInt16 (= 0)
4	1	Allocation Strategy Number	UInt8
5	3	Reserved	# 00 bytes
8	4	Start Logical Sector Number	UInt32
12	4	End Logical Sector Number	UInt32

[0182] As for a RENGUSUOBU allocation SUTORATEJII record (Length of Allocation Strategy Record:RBP 0), 16 is set up. As for an allocation strike RATEJII type (Allocation Strategy Type:RBP 2), 0 is set up. An allocation strike RATEJII number (Allocation Strategy Number:RBP 4) specifies a record of what position in an allocation strike RATEJII table this allocation strike RATEJII record is, and 0 will be set if this record is the first record. RIZABUDO (Reserved:RBP 5) is reserved for extension and #00 are set up. A start logical sector number (Start Logical Sector Number:RBP 8) specifies the head logical sector number of the field which arranges an allocation extent. And a logical sector number (End Logical Sector Number:RBP 12) specifies the logical sector number of the last of the field which arranges an allocation extent.

[0183] An allocation strike RATEJII type 1 allocation strike RATEJII record is recorded as shown in Table 41.

[0184]

[Table 41]

Allocation Strategy Record of Allocation Strategy Type 1

RBP	Length	Name	Contents
0	2	Length of Allocation Strategy Record	UInt16
2	2	Allocation Strategy Type	UInt16 (= 1)
4	1	Allocation Strategy Number	UInt8
5	3	Reserved	# 00 bytes
8	2	Number of Zones (=x ₁)	UInt16
10	6	Reserved	# 00 bytes
16	16x ₁	Zone Information Records	

[0185] As for a RENGUSUOBU allocation SUTORATEJII record (Length of Allocation Strategy Record:RBP 0), the length of this allocation strike RATEJII record and 16+16x₁ are set up. As for an allocation strike RATEJII type (Allocation Strategy Type:RBP 2), 1 is set up. An allocation strike RATEJII number (Allocation Strategy Number:RBP 4) specifies a record of what position in an allocation strike RATEJII table this allocation strike RATEJII record is, and 0 will be set if this record is the first record. RIZABUDO (Reserved:RBP5) is reserved for extension and #00 are set up. A NAMBAOBU zone (Number ofZones:RBP 8) specifies the number of the zone information records in an allocation strike RATEJII record. RIZABUDO (Reserved:RBP 10) is reserved for extension and #00 are set up. A number of zone information records with which zone information records (Zone Information Records:BP 16) were specified in the NAMBAOBU zone (Number of Zones:RBP 8) are set up. A zone information record is recorded as shown in Table 42.

[0186]

[Table 42]

Zone Information Record

RBP	Length	Name	Contents
0	4	Start Logical Sector Number	UInt32
4	4	End Logical Sector Number	UInt32
8	4	Length of Allocation Unit	UInt32
12	4	Reserved	# 00 bytes

[0187] A start logical sector number (Start Logical Sector Number:RBP 0) specifies the start logical sector number of this zone. And a logical sector number (End Logical Sector Number:RBP 4) specifies the last logical sector number of this zone. A length OBUA location unit (Length of Allocation Unit:RBP 8) specifies the allocation unit at the time of arranging in this zone. RIZABUDO (Reserved:RBP 12) is reserved for extension and #00 are set up.

[0188] A defect information table (Defect Information Table) records the logical sector number of the defective sector in logic bohrium. A defect information table is recorded as shown in Table 43.

[0189]

[Table 43]

Defect Information Table

BP	Length	Name	Contents
0	8	Signature	Signature
8	4	Number of Defect Sectors (=x1)	Uint32
12	4	Reserved	# 00 bytes
16	4x1	Defect Sector Addresses	bytes

[0190] As for the data-type field of a signature (Signature:BP 0), 6 is set up. NAMBAOBU defect SEKUTAZU (Number of Defect Sectors:BP 8) specifies the number of entries of the defect sector address which begins from (BP 16). RIZABUDO (Reserved:BP 12) is reserved for extension and #00 are set up. A defect sector address (Defect Sector Addresses:BP 16) specifies the logical sector number of the defect sector detected in this logic bohrium, and the value by which one entry is recorded on ** and here from Uint32 is sorted by ascending order.

[0191] An EKUSUTEN dead attribute table (Extended Attribute Table) consists of an EKUSUTENDEDDO attribute table header and an EKUSUTEN dead attribute record, as shown in drawing 18 . The number of the continuation in which the EKUSUTEN dead attribute record in an EKUSUTEN dead attribute table begins from 0, and ascending order is attached, and this number is called an EKUSUTEN dead attribute record number. The list of EKUSUTEN dead attribute records is created by setting a next record as the NEKUSUTOEKUSUTENDEDDO attribute record field, and this list is called an EKUSUTENDEDDO attribute record chain. The EKUSUTEN dead attribute record with which it is not used in an EKUSUTEN dead attribute table creates the list called FURIEKUSUTENDEDDO attribute record chain.

[0192] An EKUSUTENDEDDO attribute table header (Extended Attribute TableHeader) is recorded as shown in Table 44.

[0193]

[Table 44]

Extended Attribute Table Header

BP	Length	Name	Contents
0	8	Signature	Signature
8	2	Number of Extended Attribute Record	Uint16
10	2	First Free Extended Attribute Record	Uint16
12	4	Reserved	# 00 bytes

[0194] As for the data-type field of a signature (Signature:BP 0), 7 is set up. A NAMBAOBU EKUSUTEN dead attribute record (Number of Extended Attribute Record:BP 8) specifies the number of the EKUSUTEN dead attribute records in an EKUSUTEN dead attribute table, and is zero or less #FFF. #FFFF is set up, when a first FURIEKUSUTENDEDDO attribute record (First Free Extended Attribute Record:BP 10) points out the element of the beginning of a FURIEKUSUTENDEDDO attribute record chain and a free EKUSUTEN dead attribute record does not exist in an EKUSUTEN dead attribute table. RIZABUDO (Reserved:RBP 12) is reserved for extension

and #00 are set up.

[0195] An EKUSUTEN dead attribute record (Extended Attribute Record) is recorded as shown in Table 45.

[0196]

[Table 45]

Extended Attribute Record

RBP	Length	Name	Contents
0	2	Next Extended Attribute Record	Uint16
2	30	Extended Attribute Information	bytes

[0197] The NeXT EKUSUTENDEDDO attribute record (Next Extended AttributeRecord:RBP 0) specifies the following EKUSUTEN dead attribute record number which constitutes an EKUSUTENDEDDO attribute record chain, and #FFFF is set when this EKUSUTEN dead attribute record is the last EKUSUTEN dead attribute record.

[0198] Many of existing file systems are designed on the assumption that defective sector processing of media is performed by the layer (for example, shift processing inside a drive) located under a file system. It does not know where a defective sector is in these file systems, but although data can be accessed by the raw transfer rate of a drive in a portion without a defect, access is possible only in a transfer rate far lower than it with the portion into which shift processing is performed.

[0199] Since the estimate of each access time was not required even if the conventional computer use might require improvement in the mean access time, the problem did not have the above composition, either. However, audio, For the video use, since record reproduction of voice or the image cannot be correctly carried out if the constant-rate supply of the data cannot be carried out within fixed time, it has been necessary to be able to estimate time which requires a file system for a data access.

[0200] Then, the premise that it is not necessary to perform defective sector processing by the lower layer is introduced, and it enabled it to estimate correctly the time which requires a file system for access of data in this file system. In connection with this, with this file system, the field and the flag for the defective sector processing which was not in the conventional file system are prepared, and processing of a defective sector can carry out using this. Here, an example of a method which performs defective sector processing using the function prepared for this file system is explained.

[0201] It is either of the degrees that a defective sector is generally detected. During writing, an error occurs and a defective sector is detected by the 1st. Although writing terminated [2nd] normally, an error is detected when the portion is read immediately after writing. Although it wrote in the 3rd and read-out immediately after writing was terminated normally, an error is detected when reading through time.

[0202] In the case of the 1st and the 2nd, it reads immediately after writing, it is written in by performing operation (Write and Verify) of checking it has written in correctly, and can sometimes detect and respond.

[0203] In the case of the 3rd, it is the case generated with the obstacle by the contaminant in an optical disk, and the blemish etc. Although countermeasures perfect about this case do not exist, the possibility of a data loss can be remarkably lowered by performing multiplex writing. This file system mainly processes a defective sector using two technique, this Write and Verify and multiplex writing.

[0204] Bohrium structure is defined by a bohrium structure descriptor (Volume Structure Descriptor), a media information descriptor (Media Information Descriptor), a drive information descriptor (Drive Information Descriptor), and the EKUSUTEN dead data descriptor (Extended Data Descriptor). Correspondence to the defective sector to these information is performed as follows.

[0205] A bohrium structure descriptor, a media information descriptor, a drive information descriptor, and an EKUSUTEN dead data descriptor are managed by MIA. MIA can be certainly recorded on a non-defective sector by surely performing writeand verify in the case of record. Moreover, in consideration of the defect produced after record, MIA carries out the duplicate record of the MIA to

two places, and carries out a duplicate record to two places also about the MIA map which manages the operating condition in MIA.

[0206] Furthermore, it is slipping for every partition which constitutes this from logic bohrium defined by the bohrium managerial system. The defect management by linear replacement MENTO can be performed.

[0207] Correspondence to the defective sector of AV file system is performed as follows. When writing in AV file system descriptor, and Write and Verify is performed, it checks that it had written in correctly and writing goes wrong, AV file system writes AV file system descriptor to somewhere else, and rewrites the contents of the ROJKARUBORIUMU contents youth field. Moreover, reliability is raised by writing AV file system descriptor to two places.

[0208] When writing in the sector in MIA, and Write and Verify is performed, it checks that it had written in correctly and writing goes wrong, AV file system writes #FFF0 in the entry field of a MIA map, and performs the same sequence to the sector in another MIA. Moreover, AV file system raises reliability by writing the MIA itself to two on logic bohrium.

[0209] The defective sector detected working is registered into a defect information table, and AV file system can be prevented from using the sector from next time.

[0210] The data recorded on an allocation extent can perform Write and Verify (light and verification) operation from the demand of a transfer rate, and may perform only **** and Write (light) operation. When a defective sector is detected in any case, as an allocation extent which became independent about the portion, 10 is set as the allocation extent record status of the allocation extent record, and AV file system puts the allocation extent into a DIFEKUTIBUAROKESHON extent record chain. When a defective sector is detected in an allocation extent at the time of read-out, AV file system sets 11 to the allocation extent record status. When release of this allocation extent is performed, a defective sector is investigated and, as for the portion which is the defective sector, the allocation extent record status is registered into a DIFEKUTIBUAROKESHON extent record chain as an allocation extent of 10.

[0211] Drawing 19 is the block diagram showing the composition of the gestalt of 1 operation of the record regenerative apparatus 1 of this invention. It is equipped with an optical disk 8, and the record regenerative apparatus 1 reads the signal which records the video signal, the audio signal, and PC (Personal Computer) data which were supplied to the optical disk 8 from the outside, or is recorded on the optical disk 8, and outputs it outside.

[0212] The user I/O section 2 has the key panel 11 and LCD (Liquid Crystal Display) 12. The key panel 11 generates the signal according to operation of a user, and is made as [supply / the system-control section 5]. LCD 12 displays the information about the optical disk 8 with which the state of the record regenerative apparatus 1 or the record regenerative apparatus 1 was equipped etc. based on the signal supplied from the system-control section 5.

[0213] AV I/O section 3 has an encoder / decoders 13 and 14, and the multiplexer/demultiplexer 15, and controls an encoder / decoders 13 and 14, and the multiplexer/demultiplexer 15 based on the signal supplied from the system-control section 5. Moreover, AV I/O section 3 supplies the signal which shows the state of an encoder / decoders 13 and 14, and the multiplexer/demultiplexer 15 to the system-control section 5.

[0214] An encoder / decoder 13 compresses the video signal supplied from the outside at the time of record (encoding), outputs the video data of the predetermined method corresponding to a video signal to a multiplexer / demultiplexer 15, elongates the video data of the predetermined method supplied from the multiplexer / demultiplexer 15 at the time of reproduction (decoding), and outputs it outside. An encoder / decoder 14 compresses the audio signal supplied from the outside at the time of record (encoding), outputs the audio data of the predetermined method corresponding to an audio signal to a multiplexer / demultiplexer 15, elongates ODIODETA ** of the predetermined method supplied from the multiplexer / demultiplexer 15 at the time of reproduction (decoding), and outputs it outside.

[0215] A multiplexer / demultiplexer 15 multiplexes the predetermined video data and predetermined audio data of a method which were supplied from an encoder / decoders 13 and 14 at the time of record, and is made as [output / to the drive section 7]. Moreover, the multiplex video data and audio data

which were supplied from the drive section 7 at the time of reproduction are separated, and it is made as [output / audio data / output a video data to an encoder / decoder 13, and / to an encoder / decoder 14].

[0216] PC data I/O section 4 has an interface 16, controls an interface 16 based on the signal supplied from the system-control section 5, and outputs the signal which shows the state of an interface 16 to the system-control section 5. An interface 16 inputs PC data of a predetermined form supplied from the external personal computer (not shown) etc., changes them into the data which the drive section 7 can read, and is outputted to the drive section 7. The interface 16 is made as [output / are a predetermined form and / again / the data supplied from the drive section 7 / to an external personal computer etc.].

[0217] the system-control section 5 -- the user I/O section 2, AV I/O section 3, PC data I/O section 4, and the file management section 6 -- based on each state, it is made as [control / the user I/O section 2, AV I/O section 3, PC data I/O section 4 and the file management section 6]

[0218] Based on the signal from the system-control section 5, the file management section 6 controls the drive section 7, and is made as [supply / the signal according to the state of the drive section 7 / to the system-control section 5].

[0219] The drive section 7 is made as [read / a signal / have a buffer 17, the ECC circuit 18, the modulation/demodulator circuit 19, and pickup 20, based on the signal from the file management section 6, operate a buffer 17, the ECC circuit 18, the modulation/demodulator circuit 19, and pickup 20, and record a signal on an optical disk 8, or / from an optical disk 8].

[0220] A buffer 17 memorizes temporarily the data which the data supplied from AV I/O section 3 or PC data I/O section 4 were memorized temporarily, data outputted data to the ECC (Error Correction Code) circuit 18 so that there might be no way piece, and were supplied from the ECC circuit 18, and it is made as [supply / AV I/O section 3 or PC data I/O section 4] so that data may not have a way piece.

[0221] The ECC circuit 18 is made as [output / to a buffer 17], after carrying out error correction of the data which added ECC to the data supplied from the buffer 17, and outputted to the modulation / demodulator circuit 19, and were supplied from the modulation / demodulator circuit 19 based on ECC.

[0222] A modulation / demodulator circuit 19 modulates the data supplied from the ECC circuit 18 to a predetermined method, outputs them to pickup 20, and is made as [output / get over based on a predetermined method and / the data supplied from pickup 20 / to the ECC circuit 18].

[0223] Pickup 20 reads the data which recorded data on the optical disk 8 with which the record regenerative apparatus 1 was equipped based on the data supplied from the modulation / demodulator circuit 19, or were recorded on the optical disk 8, and is made as [output / to a modulation / demodulator circuit 19].

[0224] Drawing 20 is drawing showing the amount of the data currently recorded on the buffer 17 at the time of reproduction, and the relation of the speed of the data written in a buffer 17. The read-out speed R_{out} of the data outputted from a buffer 17 is controlled to become the constant value beyond a predetermined value, in order for an encoder / decoders 13 and 14 not to carry out the way piece of the output of a signal. When having read the sector on which the predetermined file of an optical disk 8 is recorded, the data drawing speed of the data supplied to a buffer 17 becomes the fixed value R_{in} , as shown in drawing 20 (B). On the other hand, data drawing speed is set to 0, while pickup 20 is moving between the trucks of an optical disk 8, or while waiting for rotation of an optical disk 8 until a predetermined sector comes to the position which can read pickup 20 (during the time T_s of drawing 20 (B)).

[0225] for this reason, since the amount of the data currently recorded on the buffer 17 becomes [being read and carried out at the read-out speed R_{out} , and] when the data drawing speed to a buffer 17 is set to 0, as shown in drawing 20 (A), it decreases rapidly the amount of data which can memorize a buffer 17 does not have data writing during the predetermined period -- ** -- read-out of data is determined by R_{in} and the read-out speed of data so that there may be no way piece

[0226] Drawing 21 is drawing explaining the composition of the file currently recorded on the optical disk 8. A block is what divided the entire disk into the equal size, and the inside of a block is continuation physically, and a data transfer is performed at the rate of R_{in} within a block. The data of a file are recorded on 1 or two or more blocks. Therefore, a block divides into the block with which the

part or all the data of a file are recorded, or the block with which the data of a file are not recorded. When the amount of data of the file currently recorded on the block is smaller than the size of a block, as for the block in front of the file, data are recorded on the all.

[0227] Drawing 22 is drawing showing the amount of the data memorized by the composition and the buffer 17 of a file. Drawing 22 (A) is drawing explaining the file currently recorded on the block. As for the block 31, the data of a file are recorded on the all. As for the block 32 which follows block 31, the data of a file are recorded on the part. As for the block 33, the data of a file are recorded on the all. As for the block 34 which follows block 33, the data of a file are recorded on the part.

[0228] Drawing 22 (B) is drawing showing the drawing speed to the buffer 17 when reading the block shown in drawing 22 (A). When reading block 31, since the block 31 is continuing physically, the drawing speed to a buffer 17 turns into constant speed of R_{in} . When similarly reading block 32 and reading block 33, and when reading block 34, the drawing speed to a buffer 17 turns into constant speed of R_{in} .

[0229] If it is not continuing since the block 31 and the block 32 are not necessarily continuing physically when ending read-out of block 31 and reading block 32 next, pickup 20 waits for rotation of an optical disk 8 until it moves between the trucks of an optical disk 8 or a predetermined sector comes to the position which can read pickup 20. For this reason, the period $Ts1$ when the drawing speed to a buffer 17 is set to 0 exists. The period $Ts2$ when the drawing speed to a buffer 17 is set to 0 when similarly ending read-out of block 32 and reading block 33 next exists, and when ending read-out of block 33 and reading block 34 next, the period $Ts3$ when the drawing speed to a buffer 17 is set to 0 exists.

[0230] Drawing 22 (C) is drawing showing the data read-out speed from a buffer 17. Data read-out speed is the always fixed value R_{out} . Drawing 22 (D) is drawing showing the amount of the data memorized by the buffer 17. Like the case where it is shown in drawing 20 (A), since the amount of the data currently recorded on the buffer 17 serves as only read-out when the amount of data of a buffer 17 reads with drawing speed R_{in} , it increases at the speed corresponding to the difference of speed R_{out} and the data drawing speed to a buffer 17 is set to 0, it decreases rapidly. In order that especially the amount of the data currently recorded on the buffer 17 when the data drawing speed after reading the block 32 and block 34 with which the data of a file are recorded only on the part is set to 0 may decrease greatly, in order for a buffer 17 to prevent an underflow, the storage capacity more than predetermined is needed.

[0231] Drawing 23 is drawing explaining other examples of composition of the file currently recorded on the optical disk 8. With this composition, the block with which the data of a file are recorded on the part or all is surely made as [record / the data of a file / or more / of a block / on 1/2].

[0232] Drawing 24 is drawing showing change of the amount of the data of the buffer 17 in the case of being constituted as a file shows drawing 23. Drawing 24 (A) is drawing explaining the file currently recorded on the block. As for the block 51 or the block 54, the file is recorded or more [the] on 1/2 as mentioned above.

[0233] Drawing 24 (B) is drawing showing the drawing speed to the buffer 17 when reading the block shown in drawing 24 (A). When reading block 51, since the block 51 is continuing physically, the drawing speed to a buffer 17 turns into constant speed of R_{in} . When similarly reading block 52 and reading block 53, and when reading block 54, the drawing speed to a buffer 17 turns into constant speed of R_{in} .

[0234] If the block has estranged physically when ending read-out of block 51 and reading block 52 next, the period $Ts4$ when the drawing speed to a buffer 17 is set to 0 exists. The period $Ts5$ when the drawing speed to a buffer 17 is set to 0 when similarly ending read-out of block 52 and reading block 53 next exists, and when ending read-out of block 53 and reading block 54 next, the period $Ts6$ when the drawing speed to a buffer 17 is set to 0 exists.

[0235] Drawing 24 (C) is drawing showing the data read-out speed from a buffer 17. Data read-out speed is the always fixed value R_{out} . Drawing 24 (D) is drawing showing change of the amount of the data memorized by the buffer 17. When the data drawing speed to a buffer 17 is set to 0, the amount of

the data currently recorded on the buffer 17 decreases rapidly. There is less possibility that the amount of the data currently recorded on the buffer 17 will approach 0 since the block 51, the block 52, the block 53, and the block 54 are recording the data more than a constant rate (1/2) as compared with the case of drawing 22 (D) than the case where it is shown in drawing 22 (D).

[0236] Drawing 25 is drawing explaining processing of record to the block of the file of the file management section 6. As shown in drawing 25 (A), the data of a file are already recorded on block 71 or 73, and processing in case the file 75 of the amount of data smaller than 1/2 of block 74 is recorded on block 74 is newly explained. As shown in drawing 25 (B), the file memorized by the block 73 leaves a part for the first portion 81 which occupies 1/2 of block 73, and is divided, and a portion 82 is moved to the head of block 74 in the second half. A file 75 is recorded following the second half portion 82 of block 74.

[0237] As mentioned above, as for the block with which a part or all of a file is recorded, a file is recorded or more [of a block] on 1/2.

[0238] When the above processing is summarized, it comes to be shown in the flow chart of drawing 26. That is, when it judges that the amount of data which judges and records whether the amount of data to record is less than [of a block] 1/2 is less than [of a block] 1/2, the file management section 6 progresses to Step S32, divides one half behind the last block of data, and is made to record them on the following block in Step S31. In Step S33, the file management section 6 records the data of less than 1/2 amount of a block on the block.

[0239] In Step S34, when it judges whether all data were recorded and is judged with recording no data, the file management section 6 returns to Step S31, and repeats processing.

[0240] When it judges that the amount of data to record is not less than [of a block] 1/2 in Step S31, it progresses to Step S35, and the amount of data which the file management section 6 records progresses to Step S36, when it judges that the amount of data which judges and records whether it is the following by 1 block is not the following by 1 block. In Step S36, the file management section 6 records the data for 1 block, and progresses to Step S34.

[0241] In Step S35, when judged with the amount of data to record being the following by 1 block, it progresses to Step S37, and the file management section 6 records the data on 1 block, and progresses to Step S34.

[0242] In Step S34, when judged with having recorded all data, processing is ended.

[0243] Drawing 27 is drawing explaining processing of division of the recorded file to a block. As shown in drawing 27 (A), one file is recorded on block 91 or 93, and the processing in the case of dividing this file into the file from the starting point of block 91 to the dividing point (ahead located from one half of blocks 92) of block 92 and the file from the dividing point of block 92 to the terminal point of block 93 is explained. As shown in drawing 27 (B), the data of the block 91 with which the portion in front of the portion 95 from the starting point of block 92 to a dividing point was memorized are divided into two, and a portion 94 is moved to block 92 the second half. A part for the first portion 95 of block 92 is stored the second half in which it was moved to the block 92 following a portion 94. On the other hand, the portion 96 from the dividing point of block 92 to a terminal point is stored in the new block 101.

[0244] Drawing 28 is drawing explaining other examples of processing of division of a file. As shown in drawing 28 (A), the processing in the case of dividing one file currently recorded on block 111 or 114 in [dividing] that it is located before one half of the positions of block 112 is explained.

[0245] As shown in drawing 28 (B), if the field of the size which can record the portion 115 from the starting point of block 112 to a dividing point is in block 111, a portion 115 will be recorded on it following the file on which the block 111 is already recorded. The portion 116 from the starting point of block 112 to the last of data is moved to a position from the starting point of block 112. 2 ***s of the data of the block 113 recorded on all ranges are carried out, it is moved to block 112 and a part for the first portion 117 is recorded following the portion 116 of block 112. The second half portion 118 of block 113 is moved to a position from the starting point of block 113.

[0246] Drawing 29 is drawing explaining the example of processing from which division of the recorded

file to a block differs further. As shown in drawing 29 (A), the processing in the case of dividing the midpoint of block 122 for one file currently recorded on block 121 or 123 as a dividing point is explained. As shown in drawing 29 (B), the portion 124 from the dividing point of block 122 to the last of data is stored in the head of the new block 131. 2 ****s of the files of the block 123 recorded on all ranges are carried out, a part for the first portion 125 is stored in block 131 following a portion 124, and a portion 126 is moved to the head of block 123 in the second half.

[0247] As mentioned above, even if a file is divided, as for a block, a file is recorded or more [the] on 1/2.

[0248] The size of the data from the starting point of a block to a dividing point shown in drawing 27 is less than [of the size of a block] 1/2, and processing of division of a file in case the size of back data is 1/2 or more [of the size of a block] comes to be shown in the flow chart of drawing 30 from a dividing point. That is, in Step S41, the file management section 6 moves back data to a new block from the dividing point of the block with a dividing point. In Step S42, the file management section 6 moves the predetermined data of the block in front of the block with a dividing point to a position from the starting point of the block with a dividing point, and moves data behind the data from the starting point of the block with a dividing point to a dividing point.

[0249] The size of the opening in front of the block with the dividing point shown in drawing 28 is more than a size of the data from the starting point of the block with a dividing point to a dividing point, and processing of division of a file in case the size of back data is less than [of the size of a block] 1/2 comes to be shown in the flow chart of drawing 31 from a dividing point. In Step S51, the file management section 6 moves data to the opening of the block in front of the block with a dividing point from the starting point of a block of the block with a dividing point to a dividing point. In Step S52, the file management section 6 moves the predetermined data of the block immediately after the block with a dividing point behind the data of the block with a dividing point.

[0250] The size of the data from the starting point of a block to a dividing point shown in drawing 29 is 1/2 or more [of the size of a block], and processing of division of a file in case the size of back data is less than [of the size of a block] 1/2 comes to be shown in the flow chart of drawing 32 from a dividing point. In Step S61, the file management section 6 moves back data to a new block from a dividing point. In Step S62, the file management section 6 moves the predetermined data of the block immediately after the block with a dividing point to the position behind the data of a new block.

[0251] Although based on whether the size of the data from the starting point of a block to a dividing point is 1/2 or more [of the size of a block] above, it is good on the basis of $(n-1) / n$ ($n=2, 3, 4$ and $5, \dots$).

[0252] Drawing 33 is continuous drawing where the free area of three blocks explains 1 blocks or more in all of processings of compression of the free area of a block in a certain case. As shown in drawing 33 (A), there are block 141 or 1 blocks or more in all of free areas of 143. The contents memorized by the block 142 are divided into the portion 144 of the same size as the size of the free area of block 141, and the remaining portion 145.

[0253] As shown in drawing 33 (B), the portion 144 of block 142 is moved to the free area of block 141. It is moved to the head of block 142, and the data 146 of block 143 are moved to block 142, and the portion 145 of block 142 is stored after a portion 145. Block 143 becomes an opening.

[0254] Thus, block 141 or the free area of 142 can be lessened, and block 143 can be made into an opening.

[0255] If the above processing is summarized, it will become the flow chart of drawing 34. That is, in Step S71, when the sum total of the opening of three blocks judges whether it is 1 blocks or more and is judged as the sum total of the opening of three blocks being 1 blocks or more, the file management section 6 progresses to Step S72, and moves the data with which the opening is equivalent to the opening of a top block from a middle block. In Step S73, from the last block, the file management section 6 moves the data with which the opening is equivalent to the opening of a middle block, and ends processing.

[0256] In Step S71, when it judges that the sum total of the opening of three blocks is not 1 blocks or

more, processing is ended.

[0257] As mentioned above, since the period when a file is recorded or more [of a block] on 1/2 in, and drawing speed is set to 0 is distributed, as for the capacity of a buffer 17, an output does not have [the block with which a part or all of a file is recorded] a way piece at least.

[0258] In addition, as an offer medium which provides a user with the computer program which performs processing which was described above, communication media, such as a network besides record media, such as a magnetic disk, CD-ROM, and solid-state memory, and a satellite, can be used.

[0259]

[Effect of the Invention] Since according to the record regenerative apparatus according to claim 1, the file management method according to claim 3, and the offer medium according to claim 4 the file of AV data is recorded and management information was recorded on at least two places of logic bohrium, an individual can do record reproduction of compression video and the compression sound signal easily by domestic.

[0260] According to a record regenerative apparatus according to claim 5, the file management method according to claim 9, and the offer medium according to claim 10 Set up the length of the unit of the information which records and records the unit of recordable information to a disk-like record medium, and it corresponds to the file recorded on a disk-like record medium. Since the length of the unit adopted when recording the file among the units of the set-up length was discriminated, an individual can do record reproduction of compression video and the compression sound signal easily by domestic.

[Translation done.]